My Career Along the Frontiers of Technology

By Bill Hayhow

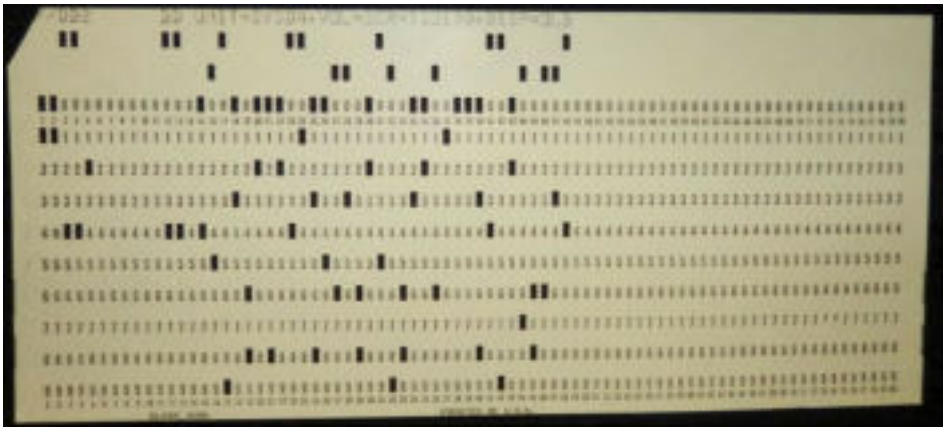
Career - 1. Early Programming

I came of age in a time before computers were in the public domain. When I was in high school, there were 'mainframe' computers and 'mini' computers (not that mini), found only in huge, climate-controlled, massively wired computer rooms within governments, universities and major companies. Access to these computers was possible via remote 'terminals', including one at my high school, which was connected via phone line to the mainframe at Michigan State University. Our terminal, a device the size of a washing machine with a built-in keyboard, was housed in a small glass-walled room, and used only by a few nerdy braniacs at my school. Despite being somewhat of a nerdy braniac myself, it never occurred to me to find out what they did in that room.

As a freshman at the University of Michigan (1978), I investigated the world of computers by enrolling in a computer programming class. By naivete, I missed the first two class sessions, prompting the professor to insist that I drop the class (because I had fallen too far behind). However, after hearing that I had shown up twice at 7:00**AM**, he admired my resolve and allowed me to remain in his 7:00**PM** class. I excelled in the class, and he and I became close.

In the days of the mainframe computer, it was challenging to interact with the computer. I did my work at the North Computing Center, which was a couple of miles from my college dorm (South Quad). Upperclassmen were granted access to 'teletype terminals', via which you typed code and instructions and received results from the computer interactively, with your own typing and the computer's responses printed onto a continuous supply of fanfold 'green bar' paper (there was no screen/monitor). Freshmen were not offered this luxury - instead we had to use computer 'punch' cards, onto which we punched our computer instructions as patterns of rectangular holes on each card.

The keypunch room offered dozens of clattering, piano-sized machines (probably the IBM 029 Keypunch). To type the many lines of a typical computer program, you put a stack of cards into the hopper at the top right of the machine. Like coins in an old-fashioned slot machine, the machine draws the cards from the hopper one at a time, across the front of the machine, and into position for punching. You then type a line of code, and press a 'release' button to move the completed card into a bin at the top left of the machine and load another blank card into position for punching.



Each card represents a single line of code (maximum 80 characters per card) so a lengthy program consumes hundreds, possibly thousands, of cards. Each vertical column on the card represents a single character, with the pattern of holes punched in that column indicating a specific character. Challenge number one was typing VERY carefully. Because the characters are rendered on the card via holes, there can be no backspace function - once the holes are in the card, there is no going back. Whether you mistype the first character on a card or the character in column 79, the card is ruined, and

you have to retype the entire card. As deadlines neared, and the hour grew late, the keypunch room resonated with cursing, the floor littered with mistyped cards flung in frustration.

Before ever setting foot in the computing center, I hand-wrote my code, or rather my best guess as to the code needed to address the project assigned by the professor. When hand-writing code, you don't get to see any results that might expose errors in coding or the logic of your solution. But it was essential to have the bulk of your thought work done before working with the computer, so I hand-wrote my code before heading to the keypunch machine.

When done with the typing, you took your stack of punched cards to the 'input' window, where a worker accepted your cards and assigned you a job number. The worker loaded the cards into a card reader, which rifled through the stack of cards like a currency counting machine at a bank. A black-and-white monitor showed the status of the jobs in process and let you know when your output was ready (at the 'output' window). At busy times in the computing center, you might wait over 10 minutes before your job was done. Then came the magic moment when you could see the brilliance of your program or, more likely, the calamity of your work.

It was not until you received the output (printed on 'green bar' paper) that you could assess your typing accuracy. Invariably, there were typing errors in the early versions of your punched cards. Your output might indicate an error on line 37 of your program, so you headed back to the punch card machine. Sometimes an error might be a simple typing error that you corrected by retyping the card and replacing it in your stack. More often, the error was a logic error, requiring rethinking your solution and adding or replacing several cards. Sadly, the computer did not identify all the errors it encountered, but only the first one it encountered. As a result, there was much scurrying between the keypunch room, the input window and the output window.

Challenge number two was that this process was frustratingly slow. As final projects came due at the end of the semester, the computing center was so busy that you might have to wait in line for a keypunch machine, even if you only needed to type a single card. As a result, I often started my work sessions late at night, just to have quicker access to the machines. You might encounter a line at the input window, and lengthy waits to get your output. Sometimes my mind was dialed-in, quickly coming up with effective solutions and quality code. More often, I slogged through the debugging process, 20 minutes at a time, finding an error, devising a solution, typing the cards, submitting my job, and getting new results.

When the computing center was busy (most of the time), it was full of beleaguered students, dragging backpacks and boxes of punch cards in and out of the keypunch room. Cards came 2000 to a box and a full box weighed 10+ pounds. A small project might require only a few hundred cards, so you might carry them rubber-banded in a stack. For a larger project, you stored the cards in the box. In either case, the cards went with you back and forth from the dorm to the computing center and everywhere within it. More than a few students suffered the disaster of dropping a stack of cards, a spectacle far worse than 52-card pickup, in that you had to rotate the cards to the correct orientation and sort them into the correct order. Many tears were shed at the computing center.

As a sophomore, I finally gained access to the teletype terminals - specifically, the DECWriter II terminal. This was a vast improvement over punch cards, foremost because you no longer needed punch cards. While you might have to wait in line for a terminal, once you were there, you could do all of your work without leaving the terminal. You entered your code via the terminal and then you edited as needed via commands entered on the terminal. You submitted your job directly from the terminal and the computer returned results directly to the terminal. All the input and output were done via a continuous stream of fanfold paper, loaded from a box beneath the terminal, and fed through a dot-matrix printer within the terminal. If the output from your program was extensive, the printing took quite some time as the maximum speed of the printer was 30 characters per second. Dot matrix printers are loud, especially when you are sitting in an area surrounded by dozens of them, but still the terminal area was much quieter than the keypunch room.

Adventure

Early in my foray into computing at U of M, my instructor gave me a hint about how you might have some fun with the campus mainframe. He didn't offer any details or instructions, only the cryptic command: "RUN ADV" (or something like that). It took me some time to find the right sequence of commands such that "RUN ADV" did anything, but when I did, the game greeted me with the response "Welcome to ADVENTURE!". "Adventure", formally named "Colossal Cave Adventure", was the first text-based adventure game, and for the next several months, it consumed many hours of my time.

Upon starting the game, the computer offers (as typed onto the fanfold paper), "YOU ARE STANDING AT THE END OF A ROAD BEFORE A SMALL BRICK BUILDING. AROUND YOU IS A FOREST. A SMALL STREAM FLOWS OUT OF THE BUILDING AND DOWN A GULLY."

You then type commands to navigate the world of the cave and interact with a variety of creatures and objects you encounter along the way. The game offers only minimal instructions, so you do much of your exploration via guesswork. The cave is seemingly vast and complex, with vexing puzzles to solve throughout the game. To facilitate my explorations, I drew many pages of maps, both to map out the places within the cave and to document how to solve the puzzles presented in each place.

Several of the puzzles are basically unsolvable, as confirmed by fellow students and the instructor, so we shared tips on how to solve them. For example, deep within the caves, you encounter a dragon which blocks access to a passageway. I tried every command I could think of to get past it, to no avail. Having given up and asked for help, I discovered you must command, "KILL DRAGON". The game then answers, "WITH WHAT? YOUR BARE HANDS?" To which you must reply, "YES" to receive this response, "CONGRATULATIONS! YOU HAVE JUST VANQUISHED A DRAGON WITH YOUR BARE HANDS! (UNBELIEVABLE, ISN'T IT?)"

I must now discuss the concept of account funding - because Adventure beguiled me into abusing it terribly. To access the university's computer, you needed an account, and at the start of each semester, the university added a limited amount of funding to your account, to 'pay' for your use of the computer. Every contact you had with the computer represented a cost which was deducted from your account. This included time spent on a terminal and also the computer processing time for each program you submitted to run on the computer.

Our instructor warned us repeatedly to be careful with our computer usage and threatened that our grades would suffer if our account balances ran out before completing our projects. In theory, the allotments were more than sufficient to complete the work for a given class. However, some people weren't as efficient at debugging their software projects and used more computer time than others. And some people (like me) used some of their allotments for extracurricular activities like Adventure. Fortunately for me, my instructor appreciated my work, so he supplemented my allotment a few times (after giving me well-deserved grief).

I started playing Adventure as a freshman, but as a punch-card-only student at the computing center, I could not play there. Instead, I discovered my dorm housed a terminal which was connected to the mainframe, just like the terminals at the computing center. I asked about it at the front desk of the dorm and, one after another, the student workers there had no idea what I was talking about. Finally, after a few days of asking, I encountered a more veteran staff member who knew about the terminal. She dug around under the front desk and presented me with a key chained to a large dongle reading 'terminal room' and a room number like 'B116'. She told me I could borrow the key whenever I wanted, but to lock up the room and return the key when I finished. Key in hand, I wandered the maze of hallways in the poorly lit basement, and finally found a door marked 'B116'.

The terminal room was tiny, perhaps 8' by 10', dimly lit by faltering fluorescents, with nothing in the room other than the terminal and a chair. The terminal was like a steampunk typewriter, a hulking antique of a machine. In my research, I haven't found an image that fully matches my recollection of the machine, but it was probably a World War II era 'teleprinter', re-purposed as a computer terminal. It was a mechanical marvel, slowly and loudly hammering characters through an ink ribbon onto fanfold paper. Printing at something like 100 words per minute, it might take a minute or more for the machine to type out a lengthy response from the computer.

Despite my dorm housing over 1000 students, I was apparently the only one who made use of the terminal room. Other than a few of my friends, I may have been the only who knew of the terminal room. It was much more efficient for me to do my computer classwork at the computing center, so I rarely used the terminal room for programming. But it worked well enough for playing Adventure. I took drinks and snacks (and my Adventure maps) down to the terminal room and explored Adventure for hours at a time, sometimes playing until I ran out of my computer allotment, at which point the computer refused further interaction. Yes, this sounds a bit sad and anti-social, but I was not unlike today's teenagers with their Xboxes and PlayStations.

Pizza Software

I took two or three programming classes at U of M, and clearly programming was a good match for me. I can think algorithmically (that is, how to build a system within the framework of a programming language), and I have the perseverance to write code and solve inscrutable problems for many hours at

a time. However, by my third semester, I struggled to imagine what career I might pursue and how college might help me pursue it. The concept of computer programming as a career was fairly obscure back then, and it did not seem to interest me at that point in my life.

A high school buddy of mine (Bruce) exited college after two semesters, taking a manager-trainee job with up-and-coming Domino's Pizza, very early in its expansion around the world. Domino's promised an exciting pathway through store management and on to franchising, with many franchisees becoming millionaires within a few years after starting with the company. This sounded more productive to me than did college, so after my third semester, I quit college to join Bruce at Domino's. We started our training in Michigan, and then moved to Texas when Domino's began rapid expansion there. Both of us became store managers, and later we followed Domino's expansion to the east coast (he to Maryland and me to Connecticut), where eventually we became area supervisors, overseeing the operations of several Domino's stores.

I was an area supervisor for only a few months. I missed my family and friends back in Michigan and I disliked my job, in part due to the long and late hours and in part because I was not well-suited to the role of supervisor. I am much more of a doer than an overseer. Another factor leading me to leave Domino's was that I finally came to identify a career path I might enjoy, and that path would require college. Having worked in the business world for a couple of years, I decided to get a business degree and hopefully, computer expertise too.

After two-and-a-half years with Domino's, I quit my job, packed up my stuff, and moved back to my parents' house in Michigan. I was fortunate that they welcomed me back with open arms, given that they had not been pleased when I left college. At that time, nearly all computer instruction was offered within engineering schools, which required extensive study in science and math. Only a few colleges offered computer-oriented majors within their business schools, so I had few options. I enrolled at Central Michigan University (CMU), majoring in accounting and 'management information systems'.

The world of computers had changed dramatically between 1978 and 1982, primarily due to the birth of the personal computer. In addition to the campus mainframe, the business school created a personal computer lab housing a couple dozen IBM PCs. Despite having only three semesters of study in computer programming, and with those classes over two years behind me, I took to my new computer curriculum as if I already knew what I was doing. I quickly became the go-to guy in the PC lab, sometimes even helping the instructors figure things out.

For Christmas my first year, I requested a 'computer terminal' and 'modem', devices which allowed me to access the CMU mainframe from my apartment. Most students had no choice but to go to the computing center to access terminals to do their programming work. Especially near the end of a semester, the computing center was so busy that you might have to wait hours to access a terminal. Once I had my own terminal at home, I no longer had to venture out, nor did I have to wait in any lines, and I could enjoy more college life than my brethren at the computing center.

Not long after I abandoned Domino's, so too did my friend Bruce. However, he had a different path in mind. Rather than become a Domino's franchisee, and thereby work within the Domino's ecosystem (and pay them substantial royalties), Bruce created his own pizza delivery company: Stopwatch Pizza. He returned to Michigan and set to work building his brand, his recipes, his first store and his business. The first store was in our hometown, East Lansing, and served Michigan State University (MSU) and environs, a lucrative pizza market.

Bruce was also interested in computers, and he and I researched what computer software might be available to help his business. We found a variety of generic business systems for accounting, inventory, payroll, etc. However, all the systems were prohibitively expensive and required extensive customization. Knowing how well I knew the pizza business, Bruce asked if I could write the needed software. I don't recall why either of us thought I could do it, but somehow we agreed that he would buy a personal computer and pay me to write pizza software to run on it.

In April 1983, he bought the state-of-the-art Victor 9000 computer for a bit more than $4,000. The Victor was far superior to the nascent IBM PC, with higher screen resolution, more memory and greater floppy drive storage. The IBM PC had a single 320KB drive and no hard drive. You inserted the boot disk to boot the computer, then replaced the boot disk with a program disk to run a program like WordStar (for word processing). Typically, you needed to swap to a third disk to store data. There was a lot of floppy disk swapping back then. The Victor had dual 600KB disk drives which prevented most of the disk swapping that was required on the IBM PC. We could boot the computer and run software from one disk, and store data on the other disk, no swapping needed.

I set up the Victor in my apartment and started learning the Basic programming language. Soon after, with bravado undeserved, I started writing pizza store software. It took a few months for me to gain enough programming sophistication to develop business-ready software, but I soon demonstrated useful results to Bruce.

Stopwatch Pizza opened in the summer of 1983, to be ready to serve MSU students upon their return to campus. I don't recall when I had software ready to use in the store, but it was probably in the fall of 1983. Software delivery was a major milestone in the project, both for its impact on the business, but also because for the software to be used in the store, the computer had to reside at the store. This required that I work remotely, a tricky concept back then. I found an obscure technique to allow me to access the Victor using my terminal and modem to connect via phone line to another modem connected to the Victor. Often, I needed to call Bruce to have him swap disks, or manage settings, or reboot the computer. Sometimes I drove to the store (an hour away) to work on the system directly. It wasn't always graceful, but we made it work.

Whether or not advisable, I sat in the back of my classes and wrote software by hand into a notebook. Back at home, I typed my hand-written code into the Victor and proceeded from there on the computer. I wrote programs to track sales, manage inventory, and even process payroll. One of my better blunders was with the payroll system. After I made a few changes to the software, Bruce ran payroll one week, and the system printed my name on every single check. I had introduced a bug into the system such that all the checks were made out to the first name in the employee database - which happened to be me.

Bruce paid me handsomely for my work. Better yet, my work for Stopwatch Pizza played a significant role in my post-college job search. Not only did I have a college degree, I had real experience writing business software. I'm sure that Price Waterhouse (my first employer after college) would not have noticed me if I did not have such credentials.

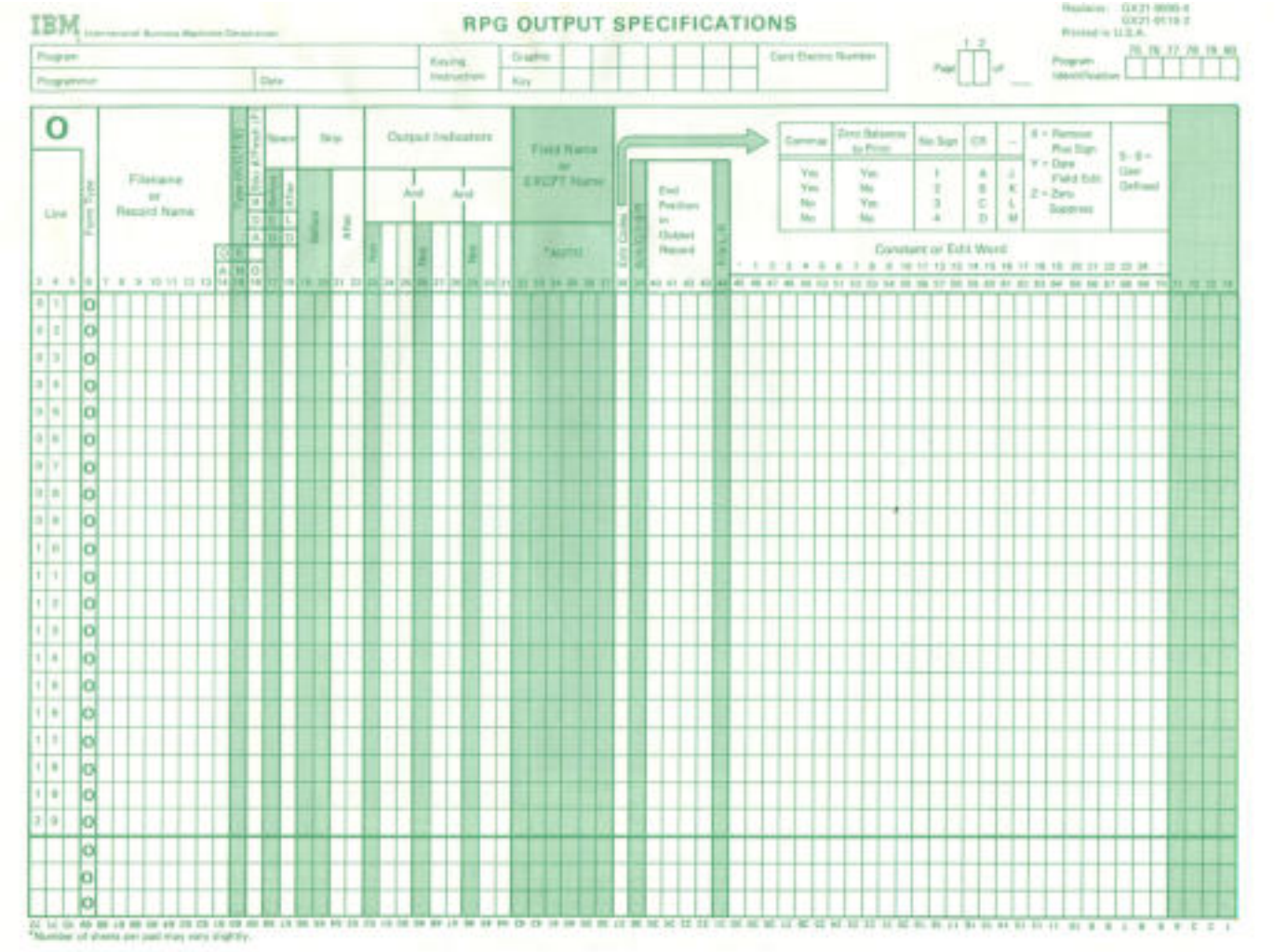
Sadly, Stopwatch Pizza closed after a couple of years. Somehow, I ended up with the Victor 9000 and over 35 years later, I still had it, 30+ pounds of retired greatness. It no longer booted up, which saddened me because I was hoping to run the software I wrote so long ago. I found a service that could copy files from my ancient floppy disks and they sent me a thumb drive containing the software. The coding was archaic and so novice as to be embarrassing. Perhaps I should have left it unseen. In 2020, I sold the Victor to a guy who restores old computers, and he got it running and showed it at the Vintage Computer Festival Midwest. I'm glad it lives on!

Programming Languages

At U of M, the beginning programming classes taught Algol W, a predecessor of the more widely used Pascal programming language, both having relatively English-like syntax. Later, throughout my career, I did most of my programming work in various versions of the Basic programming language, which also has English-like syntax. I mention 'English-like' because many other popular programming languages (e.g. COBOL, C++ and Assembly), use relatively arcane syntaxes which I find difficult to work with.

During my college years, the most popular programming language for business was COBOL and the most popular language for science/engineering was Fortran. I took classes for both COBOL and Fortran, and because I studied programming within the business school at CMU, I also took a class in RPG (Report Program Generator). Having already learned Algol W and Basic, I found COBOL to be awkward and limiting. COBOL coding requires the use of coding sheets to enforce columnar requirements based on the archaic use of punch cards (maximum 80 characters per line). To a modern eye, the COBOL language is so foreign as to be inscrutable.

Though I disliked COBOL, RPG was infinitely worse. RPG also required coding sheets, but the sheets were vastly more complex than for COBOL, with various specific columns on the sheets having specific purposes for programming. It didn't help that the professor for my RPG class was the worst college professor I ever encountered (I don't think he understood RPG any better than his students). While I aced most of my other programming classes, I did the barest minimum work for the RPG class, and I was happy to receive a 'C' rather than the failing grade I might have deserved.



Career - 2. Price Waterhouse

I graduated from Central Michigan University in December 1984, having majored in Accounting and Management Information Systems (MIS). I had the good fortune to have already lined up a job with Price Waterhouse (PW), one of the 'Big 8' accounting firms (mergers have since reduced that count). At that time, these firms had three divisions: audit, tax and consulting. My goal had been to join the consulting group within one of these elite firms, so I could focus more on information systems work than accounting. I also hoped to pursue my CPA (Certified Public Accountant certification) which would require working with the audit group and successfully completing an exhaustive exam.

My MIS degree and my experience writing pizza store software gave me opportunities beyond those offered to my fellow accounting students. The top tier of accounting graduates got job offers from the Big 8 firms, but only for the audit and tax groups. The consulting groups typically hired only top-tier MBA graduates. When submitting my resume and during on-campus interviews, I said I was primarily interested in consulting, but that I was also prepared for audit work. I received rejections from a few of the firms and job offers from a few others. But only PW considered me a candidate for their consulting group.

The consulting group in PW's Detroit office was headed by Tom Trimmer, and a significant part of their work involved a software package called FM80 (FM Financial Management for the 1980s), which had been developed by Tom's group. It was unusual for a Big 8 firm to be in the software business, but for PW it was a competitive advantage, because FM80 offered financial reporting capabilities more powerful and flexible than other software available. Besides traditional business consulting work, PW-Detroit did software development and support work and also assisted other PW offices in installing and customizing FM80 for their clients.

I had a brief on-campus interview with a PW recruiter who found me intriguing. He passed my resume along to Tom, who offered me an in-office interview. I interviewed with Tom and a half-dozen of his staff, and I'll admit that they intimidated me. But I impressed them enough that they offered me a job, and without hesitation, I took it.

In early January 1985, I moved to metro-Detroit to start work with PW. At first, I roomed with my sister Betsy at her apartment in Pontiac (technically, Bloomfield Township). Betsy had recently moved to the area for a new public relations job with GM Pontiac Division, and she had a nice one-bedroom apartment. She graciously offered me her couch while I sought housing of my own.

PW's offices were at the top of tower 400 of the Renaissance Center, the iconic glass tower complex (now housing GM headquarters) along the river in the heart of downtown Detroit. PW's consulting group occupied space on the 38th floor, with partners and managers having windowed offices along the outside of the tower and lower-level staff confined to windowless areas within. On windy days, you might feel queasy as the towers swayed in the wind. I recall one day so windy that you could see the water tilt back and forth in the toilet bowls. The first floor of tower 400 housed a movie theater, and

every afternoon the elevators delivered the smell of freshly popped corn to the 38th floor - every so often, we would feel compelled to get some.

As with any poor college student, I had neither reason nor funding to acquire the professional wardrobe I needed for my new job. As required of college seniors for interviews, I had one business suit. My father had assured me that for work, I could wear dress pants and shirts, a necktie, and one of my sport coats (aka blazer, aka dress jacket). Turns out he was wrong. While such attire was appropriate in his business environment, in my new consulting world, the only acceptable attire was a suit. I felt awkward as the only new hire wearing a sport coat, and at the not-so-gentle prodding of my elder consultants, I raced to buy more suits. I wore a suit to work every day for the next ten years, my bland uniform bedazzled only by my choice of necktie, from a lovely collection of colors and patterns, many of which still adorn my closet now 25 years later (yes dear, I'll get rid of them soon).

PW offered standard training for all new hires, an intensive several day crash course in everything PW. My training was at Indiana University in Bloomington, Indiana. I don't recall the specifics of what we learned, but I remember that college offered little preparation for the actual work of the business world. Later on, I found out much the same about PW's training - as with most professions, you learn the important things as you do them.

My (Very) Brief Excursion into Accounting

When I got back from training, I began my education about the FM80 software and the consulting work required to configure and support it. I also dabbled in the world of accounting, learning how I could earn

a CPA. To start, Tom arranged for me to participate in a few audits, the bread-and-butter work of the accounting group.

An audit is a thorough review of the accounting methods used by a company (typically audits are only required for large/public companies) to assure that financial statements are fair and accurate. My most significant audit involvement was with the Diamond Crystal Salt Company and the University of Michigan, both long-time PW clients. Every year, an army of accountants invades the client's offices to inspect their records and procedures. For an audit client, this is a significant and expensive bother, as the auditors demand the time and attention of the client's employees. Given that the audit's purpose is to uncover problems, the employees have little incentive to be helpful, so it is rather uncomfortable work for the auditors.

My role was to assist the auditors as the 'EDP auditor' (Electronic Data Processing), to review the computer systems and procedures involved in generating financial results. In 1985, computer systems were still outside the comfort zone of most auditors, and many of the client's staff as well. The EDP audit was given a fraction of the attention it deserved, and only a fraction of the attention it would receive today. For PW's audits, the EDP auditor conducted a review, and then checked a box indicating successful completion of the EDP audit. So, much as auditors are a bother to clients, the EDP auditor is a bother to auditors. The EDP audit's purpose is to uncover problems, which might stand in the way of the auditors checking the box, which might delay the 'real' work of the audit.

I arrived at both audit sites well after the audits had begun. The audit staff crowded into a conference room, an army of young, hard-working, suit-wearing accountants, engulfed in mountains of paper. They had no computers - just 10-key calculators, pads of grid-lined paper, and pencils that wrote black from one end and red from the other. I assumed I would be part of the audit team, but upon my introduction, they looked at me as if I was an alien. When they discovered that I would not be joining their paper pushing, they dismissed me as the outsider I was.

Before participating in the audits, I received perfunctory training in EDP audits, and upon arrival at the client site, the audit manager gave me a few standardized forms that PW required to be filled out as evidence of the EDP audit. Truly, I was poorly equipped for the work of EDP auditing, but it seemed this did not concern anyone. I met with a few client EDP staff, asked them questions, reviewed their procedure manuals, and filled out the forms. When done, I provided the forms to the audit manager, who checked his box and dismissed me. On my way out, I passed the open door of the conference room, seeing the overworked auditors handling their mountains of paper. A few of them glanced up at me with seeming disgust, perhaps because while I was at their same level at PW, I could waltz in and out without sharing their tedium.

After my brief audit work, I met with Tom to discuss my experience and next steps. I declared I had absolutely no interest in audit work and that I would therefore not pursue my CPA. He smiled and agreed - clearly he had known all along that I was a software guy and not an accountant, and he had encouraged me to find this out on my own. He had stacked the deck against my accountancy by having me serve as an EDP auditor, a role unliked both by the client and the other auditors. But to this day, I

am thankful to Tom because he prevented me from wasting time pursuing a CPA I would neither enjoy nor use.

My Exploration of 1985 Technology

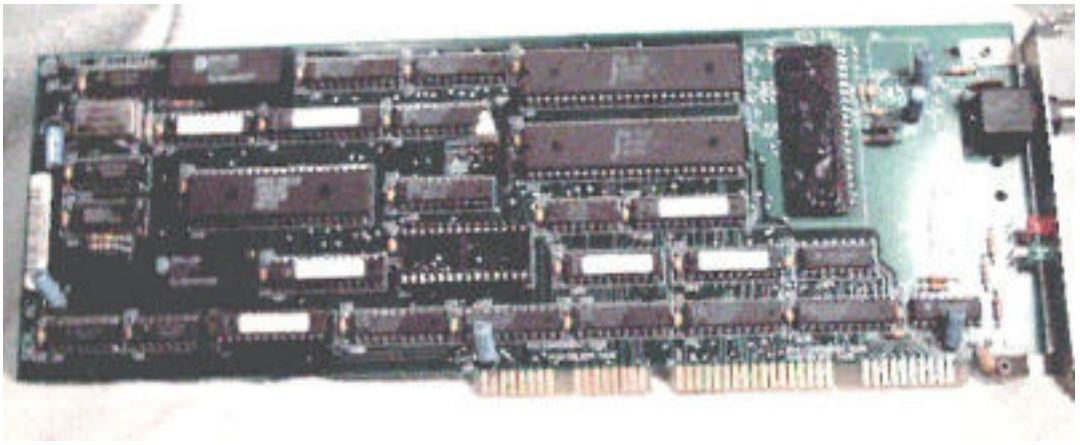
The FM80 software ran on computers running the Pick operating system, a strong competitor to Unix and other operating systems for minicomputers (a class of computers much less expensive and easier to deploy than the mainframe computers used by large companies, universities and governments). Pick was unique in 1985 (and still unique today) in ways meaningful only to programmers, but in ways giving Pick cult-like status. Pick programmers boast that Pick offers vastly more flexibility than other programming/database environments. But as the computing world shifted to GUI (Graphical User Interface - e.g. Windows) environments, Pick fell behind and ultimately faded into obscurity.

PW had a General Automation minicomputer connected to multiple 'dumb' terminals on consultants' desks around the office. A 'dumb' terminal is a computer screen and keyboard with no computing capabilities of its own. Our terminals displayed only characters (no graphics), in a single font/style, in glowing green lettering against a black background. We issued commands to the computer via the keyboard, which responded via text/messages displayed on the screen. For example, we might enter the command 'edit customer_report' to edit the programming code for the 'customer report' program. There was no 'word processing' functionality to edit code; instead we used commands to navigate the code, to add or delete lines of code, and even to edit a line of code.

While personal computers (PCs) had been used at PW for a couple of years, they were standalone devices, with minimal storage, so usage was primarily for accountants and consultants doing

spreadsheet work. Three new technologies prompted PW Detroit consultants to experiment with the PC as a platform to run the FM80 software. First, a new software product called Revelation provided a complete Pick programming environment on a PC. Second, a new generation of PC (the IBM PC AT) provided much faster processing and greater storage. Third, the Local Area Network (LAN) allowed PCs to be connected, allowing multiple PC users to work with the same set of data.

One day (within a few weeks of my hiring), Tom handed me a large red box emblazoned with 'Novell Netware', a LAN software package. Tom told me I was to become one of PW's first LAN experts. The box held a dozen floppy disks comprising the software and a stack of manuals documenting everything about configuring and managing a network. He also gave me two network cards, the electronic mechanism used to connect one computer to another via network cable. To put these massive (13") cards into perspective, consider that any Internet-enabled device you use today contains orders-of-magnitude faster wireless network functionality within a single tiny chip.



We had just received the newly released IBM PC AT, which had more than double the speed and storage of our existing IBM PC XT computers. Like teenagers with a new video game, we gathered around the beast (they weighed 25+ pounds) and marveled at its capability. It was then up to me to figure out how to network our PCs.

I read the manuals and the documentation that came with the network cards, and set into configuring a two-computer network. First, I installed the network cards into the new computer and one of our older PCs. Then, I connected the computers/cards with coax cable and booted the computers. I don't recall the details of installing the Netware software or the days-long and frustrating process of getting everything to work. But I remember my elation when I succeeded, as proven via a network game called Snipes.

Snipes is a multi-player game in which the players navigate a maze to find and kill each other. While it was crude by today's standards, it was surprisingly entertaining, and we often used it for a mental break during the workday.

Our next job was to install the Revelation software and get the FM80 software running on the PC within the Revelation environment. I don't recall how we copied the FM80 code from the minicomputer to the PC (memory sticks didn't yet exist and there was no floppy disk format compatible with both minicomputers and PCs), but we probably used a modem to connect the PC to the minicomputer via phone line. We then had to make many changes to the code to accommodate differences between Pick processing on the minicomputer and within Revelation. Within a few weeks, we solved all the issues and had FM80 running just as well in the PC/network environment as it did on the minicomputer.

After a few months, Tom told me I would join a few members of the FM80 team in attending the annual FM80 User Group conference in Tampa, Florida. Oh, and he told me I would be one of the presenters, my topic to be Local Area Networking. This was a daunting challenge for me, as I was still the rookie on the team and had only minimal knowledge of LANs. But, like it or not, Tom urged me to a podium, and with a few slides on an overhead projector, I passed on what I knew about LANs and how they could benefit the companies represented in the audience. I've since become a fan of trial-by-fire; just agree to do something daunting and give it a shot. Most of the time you'll be successful and gain confidence and expertise along the way.

Allow me to expand on the topic of 1985 technology. In 1985, writing was generally done by hand, with documents such as client correspondence hand-written onto notepads, complete with scratch-outs and editing marks throughout. When you completed work on your notepad, you (or for PW partners, your secretary) walked the pages up to the 39th floor to the 'Word Processing Department' (as indicated by the sign on the wall), whereupon you handed the pages to a word processing specialist who placed your job/request into a queue awaiting its turn to be typed by one of several typists in the department. Of course, partners received top priority, but for the rest of us, it was essential to be very, very nice to the word processing people, to make sure our jobs did not languish there all day. This was crucial because writing and word processing was an iterative process. The first typed draft of your document invariably needed modifications, due to misinterpretation of the scrawled notes you had provided. It might take several iterations to get to a client-ready document, so we were absolutely beholden to the word processing department to help us produce finished documents on time.

Given the problems caused by this word processing process, and our expanding expertise in PCs, the consulting group soon moved to PC-based word processing. Unlike the accountants, many of us in the consulting group typed on keyboards every day (writing code and documentation). For us, the issue was not the typing but the printing. The printers we used with our PCs were dot-matrix printers, which were relatively fast and inexpensive to use (and loud!), but which generated poor-quality output, far below the quality needed for client correspondence. Our solution was to equip our secretaries with daisy-wheel printers, which produced high-quality output using a technique much like a typewriter, that is bashing fully formed letters through an ink ribbon onto the page. We saved our documents on a floppy disk and took them to the secretary, who loaded the document onto her computer and printed them out. This saved us much time and angst.

Also in 1985, PW Detroit acquired its first Apple Macintosh computer. In 1985, the business world relied almost exclusively on IBM PC and compatible computers. There was little business software available for the Macintosh and it was considered useful only for creative types, for desktop publishing, for artists of all sorts. Our consulting group came to include an electrical engineering group, and they acquired a Macintosh to assist with diagramming electrical systems. The Macintosh had a Graphical User Interface (GUI), rather than the text-only interface on the IBM PC. This allowed them to draw and print electrical diagrams and eventually write graphics-based software for their clients. The rest of us eventually found ways to make use of the Macintosh as well. Using Apple's MacDraw software, we produced flowcharts and other diagrams that added pizazz to our client communications.

My work with PW and FM80 required extensive travel (see My Client Work below). Often, I would fly out Monday morning and return home Friday night (sometimes we got a reprieve and returned Thursday night instead). While we did much of our work on the clients' computers (minicomputers running the Pick operating system), it became essential that we bring our own PCs, so that our work could travel with us, and so that we could do computer work anywhere (home, office, airport, hotel). Enter the portable computer, also known as luggables, as they weighed over 30 pounds. My first portable was a Compaq Portable 286, which was equivalent to the IBM PC AT in terms of speed and storage, and which cost over $4,000, far more than a top-end laptop would cost today, and a small fortune back then.

I lugged one of these everywhere I went for a couple of years, through many miles of airport terminals, and in and out of offices and hotels and rental cars. When traveling, I carried the computer plus a jam-packed briefcase and a hanging garment bag containing suits, shirts and neckties for my work week. Yes, I have a back and shoulder problems, and undoubtedly the luggables contributed to my woes. On a positive note, the Compaq was so sturdy that it served as an acceptable stool, offering a place to sit in a crowded airport after yet another Friday night flight was delayed.

A few years later, we joyfully upgraded to the Compaq Portable III, which weighed a mere 20 pounds. Starting with the IBM PC in the early 1980s, we awaited each new generation of computer like a kid awaiting Christmas. We would overload and overtax our existing PCs to where they seemed to crawl through our computing tasks. The next generation of CPU (Central Processing Unit, aka processor) would be announced (first Intel 8086, then 80286, 80386, etc.), and we would wait impatiently for computers with the new CPU to become available. When the new PC finally arrived, a crowd would form, and we would tear open the packaging, and fire up the new computer. With joy and relief, we would watch the new computer chew through a task in a fraction of the time it took our old computers.

My Work with FM80

FM80 was developed by PW Detroit to address complicated financial reporting projects that would be difficult to address with existing software. In short, FM80 was a general ledger system, used to record the financial transactions for a company, and then to produce financial reports and statements compiled from those transactions. Unlike any existing general ledger systems, FM80 offered incredible flexibility within the chart of accounts and corporate structures created within the system. Typically, a chart of accounts was a fixed outline-like structure, with low-level accounts like 'office supplies' and 'postage' summarized to higher-level accounts like 'office expenses'. Within FM80, the hierarchy of accounts was more fluid, such that it became easy to revise financial statements by rearranging the account hierarchy. It was even possible to create multiple versions of the chart of accounts, which allowed companies to prototype changes to their financial statements.

Similarly, the departments, divisions and cost centers within a company are defined within hierarchical structures. Consider companies like GM or Microsoft. They have multiple subsidiary companies, and financial statements need to be generated for each of them, and then consolidated for the company as a whole. Often, subsidiary companies have their own subsidiaries (or 'cost centers'), each of which requires its own financial statements. For example, Microsoft sells the Xbox gaming system and related software (games) and services. Financial statements are generated for each Xbox segment (hardware, software, services), and then aggregated for Xbox as a whole, and then aggregated with other

segments (Skype, Windows phone, etc.) to produce statements for the 'Entertainment and Devices Division'. Those statements are aggregated with Microsoft's other major division (Online Services, Business Software, etc.) to produce financial statements for Microsoft as a whole.

FM80 offered flexibility in how these hierarchies were arranged, and as with the charts of accounts, it was possible to create multiple versions of the hierarchies to generate a variety of aggregated financial statements. It also made it easy to reconfigure financial statements to support corporate restructuring. As companies acquire and dispose of subsidiaries, and as their business offerings evolve, they regularly rearrange their divisions and organization (management) structures. FM80 made it relatively easy to reconfigure reporting hierarchies to match corporate restructuring. Some of our largest FM80 consulting projects involved helping companies prototype new corporate structures by building multiple financial statement hierarchies for them.

Another major project category for us was integrating all the financial transactions from throughout a company into the FM80 system. With each acquisition comes a new accounting system that must be integrated into the corporate system. Each company and subsidiary might have different department structures and accounting methodologies that needed to be homogenized into a corporate standard. We wrote fantastically complicated translation programs to take transactions from various sources throughout a company and re-categorize them to fit into the corporate standards.

Traverser

While word processing ultimately became one of the primary uses for PCs in an office, it was the spreadsheet that instigated the business use of PCs. The first spreadsheet software was Visicalc (1979), but a few years later, Lotus 123 (1983) became the industry standard. Both Visicalc and Lotus faded into history (along with most of the early PC software packages), but for several years, Lotus was ubiquitous, and remnants of its ubiquity remain to this day.

Although the Apple Macintosh offered a GUI with mouse navigation, and although you could use a mouse to interact with the text-based software on PC compatibles, most of us interacted with PCs using only the keyboard. With Lotus 123, you accessed the menu (e.g. File, Edit, View, etc.) by pressing the slash (‘/’) key, and then pressing the first letter of a menu option (e.g. ‘F’ for ‘File’) and the first letter of a submenu option (e.g. ‘O’ for ‘Open’) to issue a command to the software. Over time, we used various key sequences so often that we used them without thinking or looking at the keyboard. To ease the learning curve for new users, competitors adopted these same menu structures and key sequences. To this day, in Microsoft Excel, you can press '/FO' to access the 'FIle Open' function.

A few months before I started at PW, one of the PW Detroit consultants (Brett Boal) prototyped a spreadsheet interface for browsing the financial data in FM80. Financial spreadsheets often present financial accounts (e.g. expense categories) as rows and time periods as columns, but they can just as easily present various operating units (subsidiaries, divisions, departments) and time periods, either in rows or columns. Brett's invention (others helped inspire the software, but Brett built the prototype), offered a spreadsheet interface that allowed you to assign business dimensions to the rows and columns of a spreadsheet, and to navigate the hierarchy for each dimension based on the

organizational structures and charts of accounts configured within FM80. They named the software Traverser, based on its abilities to traverse the full extent of a company's financial information. Given that its target audience was financial analysts, who were extremely well-versed in Lotus 123, Traverser mimicked the user interface of Lotus 123, including the slash and many of the commands.

Development and support of the core FM80 system was moving to a dedicated software development team in PW's Tampa office. PW Detroit continued to do the most demanding FM80 customization work, including the development of tools like Traverser. Tom and Brett encouraged me to take on the continued development of Traverser, to free up Brett to lead our FM80 implementation and customization projects. In addition to my extensive client work and travels, I took on Traverser development as a side project, moving it from prototype to a fully realized software product we called Traverser 2.

With Traverser 2, you could start with a high-level view of a company's income statement, with top-level account categories as the rows, and the past few years as the columns. As you browse the numbers, you might see discrepancies, such as an expense category much higher this year than last. With a click, you could 'drill down' into that number to produce a new spreadsheet, showing a breakdown of the specific accounts within the expense category, or a breakdown of the expense category across the company's operating divisions. You could drill down all the way to the underlying financial transactions and back up again. You could choose the business dimensions to present in the rows and columns, choose where and how much of each hierarchy to include, and choose how to present the information within the spreadsheet. Before Traverser, such financial analysis required navigating reams of financial reports and transactions, which required great concentration, persistence, and time. Traverser made such financial analysis much easier.

My Client Work - Baldwin United, Philadelphia

My first out-of-town client was Baldwin United (BU), a vast conglomerate that grew from its start as a piano company to include dozens of subsidiary companies, including several insurance companies and S&H Green Stamps. BU collapsed in 1983, amid speculation of fraud, resulting in the biggest bankruptcy in history at the time. In 1985, the bankruptcy was being adjudicated in US Bankruptcy Court in Philadelphia, and the court authorized BU to employ an array of lawyers and consultants to explore options for restructuring the company to best satisfy creditors.

PW's role was to load the financial statements for each of the subsidiary companies into FM80 and then to generate financial reporting based on various concepts for restructuring the company. The consultants might ask us, assuming that ' we merge these three subsidiaries' and 'sell off those two subsidiaries', what would the financials look like? For me, this was an immersive lesson into the world of consulting and the intricacies of FM80.

This was also an early outlier in terms of accommodations. While our clients hosted us reasonably well on all our projects, they typically provided modest office spaces, and we stayed in modest hotels and enjoyed modest restaurant meals (neither gourmet nor fast food). For the BU project, the bankruptcy court had approved a generous per-diem (that is a daily per-person expense budget) that allowed us

fancier hotels and restaurants. My first night in Philadelphia, I stayed at the Four Seasons Hotel (not Four Seasons Landscaping, a mockingly 2020 reference). My room was spacious and elegant, with a plush white bathrobe hanging in the bathroom. After a far-better-than-I-usually-get breakfast in the opulent hotel lobby restaurant, and feeling like the highfalutin consultant I had become, I walked down the steps of the fancy hotel and then half a block away, I stepped past a homeless guy sleeping on a steam vent on the sidewalk. This inspired a quick awakening in me, that the distance from Four Seasons to homeless is not very far.

I spent a few months traveling to Philadelphia, and I enjoyed my time there greatly, although I must point out how foreign it was to a Midwesterner like me. I was used to walking the streets with my head up, making eye contact with passersby, and sometimes even offering greetings. In Philly, this is simply not done (so too for most East-coast cities). One of my favorite memories is visiting Bookbinders restaurant. My grandparents and parents used to enjoy getting a bowl of Bookbinder soup at the fabulous Drake Hotel in Chicago. Bookbinder soup, also known as snapper soup, is a rich turtle soup (yes snapping turtle, although other recipes call for red snapper fish) served with a small pitcher of sherry to adorn the soup. I was thrilled to be able to go to the Old Original Bookbinder's restaurant in Philadelphia, to eat a bowl of the original-recipe soup. In what would become a regular disappointment for me, I found that enjoying such treats alone is not so thrilling.

Another fond memory I have is of the Latham Hotel, which I assume we selected because it cost far less to stay there than at the Four Seasons. Nonetheless, I fell in love with the Latham, because it taught me to love city life. On more than one stay, I had a corner room with windows that opened, allowing the nighttime breezes to blow through the room, carrying the scents and sounds of the city. While I have little interest in living in such an urban setting, there is a compelling vitality to urban life, and I'm glad to have experienced some of it.

My Client Work - Ruan Trucking, Des Moines

In late 1985, we began a project to implement FM80 for Ruan Transportation, a large conglomerate of trucking and related transportation businesses, headquartered in Des Moines, Iowa. Unlike my experience with Baldwin-United, at Ruan, our team was sequestered in a small, dark, out-of-the-way office (or at least that's how I recall it). And compared to Philadelphia, Des Moines is barely a city at all, with a fraction of the ruckus, and city life that seemed to come to a halt at 9PM.

Our challenge with this project was to integrate the financial transactions from hundreds of subsidiary companies into the corporate FM80 system. We developed a sprawling import program, which we gave the bland name 'Source Translator', to reclassify financial transactions from each subsidiary to fit into Ruan's corporate standards. This was not exciting work. We delved deep into the minutiae of each subsidiary's accounting methodologies and implemented thousands of minor translations.

Des Moines is near the eastern edge of the Central time zone, so it gets dark early in winter. Winter there is so cold and windy that they built a vast network of interconnected indoor walkways, to allow you to traverse much of the downtown without venturing outside. We could walk from our hotel, through a shopping mall, and all the way to Ruan's offices without needing a coat.

Des Moines night life was nearly as tedious as our project work. When we worked on out-of-town projects, we typically worked late, because we wanted to complete our work, bill our hours and go home. Often, we wouldn't call it quits until after 7PM. In Des Moines, we had to watch the clock, because many of the downtown restaurants closed at 8PM (at least in the winter months), and if we worked too late, we might have trouble finding dinner. Perhaps our most enjoyable meals were at the Iowa Beef Steakhouse, which featured a wall of coolers containing massive cuts of beef - like 24 ounce sirloins and 20 ounce ribeyes. You could browse the selections, pick out your own steak, and if you wanted, you could even grill it yourself on a huge charcoal grill in the center of the dining room. I don't recall if I ever finished one of these meals, but I know I gave it my best effort.

I had one of my better programming blunders at Ruan. Our Source Translator was so massive and complicated that it took a few hours to run, so we would finish all of our programming updates and run the program overnight. One morning, we came back into the office to find it overflowing with printer paper. We had loaded a fresh box of continuous fanfold paper (2500 pages), and the program had worked long into the night, printing the same line over and over until it ran out of paper. Unfortunately, I had introduced an 'infinite loop' into our code, a bug that causes a program to run the same code endlessly, and that code included a print command. Yes, the other consultants made me clean up the mess, both the paper and my bug.

My Client Work - FHP, Newport Beach (CA)

My circumstances improved for winter 1986, with a lengthy project at Family Health Plan (FHP) in Newport Beach, California. FHP was a pioneer in the Health Maintenance Organization (HMO) industry. In the 1980s, FHP was rapidly growing, with over 300,000 members and operations in several states and (oddly) Guam. For FHP, we implemented FM80, Traverser, and some related systems to improve their financial reporting and help them manage their fast growth.

Whereas in Des Moines, we toiled long hours in a bland office, in a bland city, enduring winter's icy wrath, in California we enjoyed sunshine and fresh air and client staff suggesting that we work less and enjoy California more. We might have quit early a few times, but mostly we still worked long hours so we could finish and head home. For me in particular, the travel was becoming a problem. My wife Lisa and I had married in June 1985, a few months after I joined PW, and we moved to a nice flat in Detroit not too far from my office downtown. The problem was that I was traveling extensively and Lisa had few friends in the area, so she was unhappy with my continued absences. I had far less fun on my work travels than one might expect.

We enjoyed many perks at FHP. We worked directly for the CFO (Chief Financial Officer), and he had a large budget and an aggressive agenda. As long as we kept producing results and meeting deadlines, he was happy, and he would gladly sign off on our sizable billings, including some rather extravagant expenses. We dined well. Several times we dined at The Arches in Newport Beach, which specialized in Caesar salad prepared table-side and fried abalone (a large mollusk/snail, and a delicacy). We also worked our way through several vintages of Jordan Cabernet Sauvignon, one of the better California cabs. Yes, we gloated a bit with our fellow consultants back in Detroit, but we were more careful with our wives, fearing they would not appreciate us having such experiences without them.

On to Greater Things

PW's employment hierarchy included 5 levels - in the consulting group, these were consultant, senior consultant, manager, senior manager and partner. The general concept was that consultants should advance to managing projects and staff, then to selling additional work, and ultimately to creating and managing client relationships. PW expected you to move up a level every 2 to 3 years, and that if you were partner material, you should get there within 10 to 12 years. At that time, PW had an 'up or out' philosophy - either advance up the ladder or find employment elsewhere (and PW would often help you find your next job, hoping that you would hire PW to do work for your new company). Particularly for consultants, this philosophy was changing during my time there. While 'up or out' might work for the accounting group, it created issues in the consulting group, because it forced out those with deep area expertise.

I started as a consultant and was promoted to senior consultant after two years. I worked closely with Tom Jenks, a newly promoted senior manager. Despite our progress up the ladder, Jenks and I saw an opportunity that would lead us to leave PW. Our client work, particularly with FHP, involved financial reporting to the highest levels of management within a company. We saw that our Traverser technology allowed executive management to browse financial data in a way never before offered to them. A new software category was coming to life: the Executive Information System (EIS). Our key contact at FHP (Carl Peternell) encouraged us to explore the concept of developing and commercializing EIS software. Sadly, while writing this, I discovered that Carl died in 2012 - he was a great guy and he will be missed.

On our weekly flights from Detroit to California and back, Jenks and I discussed what software products we might produce, where and how we might sell them, and what it would take to start a software company. PW wasn't interested in being in the software business, so if we were to be in the EIS business, it would have to be via a startup company. We realized we would need help in setting up such a company, and in connecting with the executives who would eventually purchase and use the EIS software. We called Tom Trimmer to ask if he would be interested in joining us.

Early in 1987, Tom had left PW, after declining PW's second request for him to move to another PW office. PW had another philosophy, that partners should gain experience and contacts in multiple cities, so they routinely reassigned partners to different offices. Tom did not want to uproot his family, so he declined their requests. We missed having Tom as our champion - he had protected us from forces within PW that demanded we focus less on national FM80 consulting, and more on consulting projects that would dovetail with PW's accounting and tax services. As a retired PW partner, Tom had many contacts, and he had spent the past few months exploring his options. We were happy (and a bit surprised) when Tom agreed to join us as one of our founding partners.

The three of us worked out the details of our partnership, with Trimmer serving as president with a 50% stake, and Jenks and I each with 25% stakes. We named our venture Quest Management Systems, which in retrospect, seems trite and possibly pompous, but I suppose it was as good a name as any. We negotiated a partnership agreement between us and worked on financing, office space, computers, etc. It was both exciting and scary to consider leaving PW for a startup, but I was young and naïve and honored to be included as a founder, along with Trimmer and Jenks.

While it would be a bumpy ride, and never generate much financial success, I'm glad I took this leap, and I have no regrets about leaving PW. The PW partner track was demanding, both for consultants and their spouses, and I'm sure that I was not partner material. Given that PW was not interested in software development, and that software had become my passion, I don't believe I would have lasted there. Quest was an opportunity that I could not refuse.

When I left PW, my billing rate was $150 per hour. Across the 30+ years since then, despite inflation and my growing expertise, I have never charged a higher rate. It takes high rates to pay for fancy offices at the Renaissance Center and hefty partner salaries and perks. As an entrepreneur, you learn to make do with a lot less.

Membership Management System

In my spare time (though it's hard to imagine how I had any), I wrote software - yes, I had somewhat singular focus back then. My father's public relations company (Publicom) provided PR services to numerous associations who lobbied the Michigan state government on behalf of their constituents. In the 1980s, Publicom's services expanded to include association management, to help associations manage membership information and dues, events, conferences, etc. As there were few options for membership management software, I agreed to develop a PC-based membership management system for Publicom.

I developed the system using dBase III, which at the time was as ubiquitous for PC-based databases as Lotus 123 was for spreadsheets and WordPerfect for word processing. dBase included a programming language with which you could build applications that ran fully within the dBase environment. For storage, we used an intriguing device called the Bernoulli Box, which used floppy disks inside rigid cartridges the size of a thick magazine, but via Bernoulli's principles, achieved hard-disk caliber speed and capacity (20MB per disk). This worked well for Publicom, in that they could keep a separate disk/cartridge for each of their association clients. Within a couple years, Publicom implemented a suite of software from MAI Systems, sold and supported by Bickley & Associates (see later references to Craig Bickley). The MAI software included membership management capabilities and fully replaced the application I had developed. By then, I was so busy with my own software developments that I was glad to no longer support this one.

However, I never left behind the concept of membership management systems. Every organization of which I've been a member has needed a membership management system, and I've provided a solution. This has included sports teams, church and, of course, the Viking Club.

Career - 3. Quest Management Systems

Executive Information System

After less than two-and-a-half years, I left Price Waterhouse (PW) to begin my career as an entrepreneur, as co-founder of Quest Management Systems, along with my partners Tom Trimmer and Tom Jenks. From the outset, we were busy - setting up the business, moving into office space, and working on business and software development plans. We had a few clients, mostly generated from Trimmer's explorations after leaving PW. Trimmer and Jenks pursued additional consulting clients and I began creating the architecture for the EIS (Executive Information System) software that was the instigation for our venture.

Like an artist in front of a blank canvas, or an author staring at a blank page, it is both liberating and daunting to write software from scratch. Usually, software is written within the realm of an existing architecture, and you must abide many layers of rules and structures which have evolved and solidified over the life of that architecture. Starting from scratch allows you to design freely, without the limitations imposed by immutable rules and inflexible structures. However, the blank canvas stares back at you, daring you to be bold, yet mocking any deficient art. Despite my youth, my partners had faith in my software expertise, so I set into designing an EIS architecture. Trimmer and Jenks helped conceptualize the system, and I worked out the details.

We extrapolated from the architecture of PW's FM80 system, to design a new system that could represent the varied financial reporting dimensions of a business and to accommodate a wide variety of use cases. Early working titles for the system were InfoQuest and ConQuest, but we ultimately settled on ActionQuest (clearly, we were questing). I still have some of our preliminary design documents, and they show our grand ambition, and possibly pretension.

From the start, we sought investors to help fund our vision. We created a six-year strategic plan showing lofty projections for growth, revenues, and profit. The plan also showed multiple rounds of investment from venture capitalists. Our estimated (and wildly optimistic) year six valuation was over $50 million based on annual revenues of $28 million and profits of $10 million. Considering inflation from 1987 to 2020, these numbers would more than double. We pitched our plan to several investment groups, and while we attracted interest, we could not attract investment.

As a result, we were under-funded. To pay salaries and expenses, we turned to consulting services, taking on projects much like we had done at PW. As my consulting workload increased, I could devote a dwindling amount of time to software development. In our first few months in business, I developed some basic prototypes of components of the ActionQuest software, but development soon stalled. We never completed the design, nor did we develop much of the software.

My Client Work - Ford Motor Company, Detroit

Early in Quest's second year, we landed a consulting job with Ford Motor Company. PW was working with one of Ford's divisions (Climate Control) to gather extensive information about their competition, primarily to compare competitors' cost structures to Ford's. They encountered problems with recording and reporting the competitor information they were collecting - because much of the information was 'fuzzy'.

Competitor information is often anecdotal (e.g. 'there is more truck traffic at Toyota's California factory'). Numeric information is often estimated or presented in ranges, and details about the source, accuracy and trends about a number are often more valuable than the number itself. Traditional databases and spreadsheets are ill-suited to storing such 'fuzzy' information and they offer poor tools for analyzing it.

One of Trimmer's fellow partners at PW Detroit was leading the Ford project for PW. Knowing our expertise with sophisticated databases, he asked Quest to develop a system to support their project. This was a valuable project for us, both for the hefty project billings, but also for the path it set us on. Quest would soon become a pioneer and leading provider of Competitive Intelligence systems.

My Client Work - Maison Blanche, Baton Rouge

Tom Trimmer was the visionary behind the development of PW's FM80 financial management software, Tom Jenks was one of the most experienced FM80 consultants, and I was one of the most knowledgeable FM80 software developers. The three of us together, along with a couple of consultants we hired along the way, offered elite-level FM80 consulting services. At Quest, we worked on several major FM80 consulting and customization projects, sometimes helping PW with their client projects, and sometimes generating FM80 projects of our own.

We did extensive work for Maison Blanche (MB), a fast-growing chain of department stores, headquartered in Baton Rouge, Louisiana. In the late 1980s, MB acquired over 15 department stores in Florida, and they needed help with integrating and reporting their financial information.

Our work for MB was not sufficiently distinctive to merit description here. But I can offer my fond remembrances of Baton Rouge - in particular, the food. I enjoyed Po' Boys, crawfish étouffée, and gumbo. But my fondest memories are of Mike Anderson's, a legendary Louisiana seafood restaurant. When we first arrived in Baton Rouge, we asked our client cohorts where we might find good local food. They sent us to a large restaurant along the highway, with parking sufficient to accommodate the many tour buses that would unload tourists into the bland, touristy restaurant. The next day, we asked them, "where would **you** go to get a good Cajun meal?" Without pause, they said, "Mike Anderson's."

Our first time there, I followed the server's recommendation and ordered the 'Fried Seafood Platter', a massive heap of fried joy including: shrimp, oysters, crab claws, crawfish tails, stuffed crab, catfish fillets, onion rings and hushpuppies. A single order was enough food for several people, but I faced this monstrosity alone, and I made a credible effort to consume it. On another occasion, we sat at the bar for oyster night, sipping beers and watching the staff shuck bushels of fabulously fresh oysters. We ate dozens of raw oysters, adorned with a variety of toppings, including hot sauces, horseradish, and fresh lemon juice (my favorite). I'm happy to report that I had no gastrointestinal issues the next day.

My Client Work - Other Notable Experiences

While at Quest (1987 to 1993), I did work for a wide variety of clients. Over the years, our work shifted from FM80 work to more general consulting work to almost exclusively competitive intelligence work.

One of our clients was Quadrant Group, a financial services company specializing in pre-retirement planning, a service to help people better understand when and how they could retire. I built them a pre-retirement planning system, into which they could enter dozens of factors describing someone's financial situation, to generate a detailed analysis of their prospects for retirement. I bring this up here, because the two principals at Quadrant gave me a lifelong aversion to financial planners. It's trivial, but both of them maintained perfect manicures, and my impression was that they cared far more about money than their clients' well-being.

We worked on a variety of projects for Amerimark, a manufacturer of aluminum and vinyl siding (and related products). For Amerimark, I got to travel to Gnadenhutten, Ohio, a tiny town deep in the foothills of the Appalachians. The Gnadenhutten facility was fascinating - they melted down scrap aluminum (literally anything made of aluminum), purified it into slabs, pressed it into sheets, rolled it through paint, oven dried it, shaped it and cut it into pieces of siding and gutters, all as one (mostly) continuous process.

I highlight Gnadenhutten partly for the name itself (who goes to a place called Gnadenhutten?), but also to note that while some of my travels were arduous (Ohare), or even gruesome (much of New Jersey), some of my travel was delightful. I once drove from Gnadenhutten to Washington, D.C., to get to a conference. I had time, so I took the back roads rather than the highway, winding my way through the Appalachian mountains, enjoying stunning vistas with blooming dogwoods sparkling along the hillsides. To this day, I prefer back roads to the highway (ask my family).

Amerimark also had a facility in Roxboro, North Carolina, a small rural town surrounded by tobacco fields, just outside North Carolina's Research Triangle area. While I enjoyed nearby Durham (where we stayed), Roxboro offered nothing memorable other than oppressive heat and humidity. A visit from my wife Lisa and toddler son Brian happily interrupted the unpleasant tedium of this project. Lisa had rarely visited any of my remote work sites, with both of us preferring that I return home for some normalcy. I'll never forget my joy upon seeing Lisa and Brian arrive at the airport, Brian towing a suitcase full of toys down the concourse, then shouting "Daddy!" and running to me for a hug.

INCITE

Our competitive intelligence work with Ford inspired us to pursue opportunities in the competitive intelligence (CI) industry, a relatively new and unfocused practice within Fortune 500 caliber companies. For years, such companies had employed business analysts and marketing consultants to monitor market conditions and competitor activities. Due largely to technological advances, CI was becoming much more widely practiced. PCs and spreadsheets allowed analysts to more easily compile CI data. Electronic communications (e.g. email and networks) allowed people throughout a company to

more easily share CI data. We saw an opportunity and began developing general-purpose CI software for this quickly growing market.

The system we created for Ford addressed many of the primary needs of a CI project, but it was focused on the specific requirements of Ford's project. The general-purpose CI system we envisioned needed a flexible structure to adapt to the varying business dimensions of different industries and applications. A consumer goods company analyzes much different competitive factors than a manufacturer. A marketing department considers different factors than an R&D department.

We named our software product INCITE, an acronym for INovative Competitor Information TEchnology, but with the plus of invoking both 'insight' (i.e. 'understanding') and 'incite' (i.e. 'take action') along with having 'CI' in the middle of the word. When we introduced a Windows version a few years later, we simply added a 'W' to the front to create 'Wincite'. I think we were better at developing software than naming products.

INCITE was a database system organized based on 'dimensions' (e.g. Company, Location, Product, Market) and Essential Elements of Information (EEIs), which are akin to database fields. Our INCITE consulting practice guided companies to identify EEIs to help them focus on the specific competitor information that was most important for them to monitor. INCITE included functions to enter and annotate facts, figures, rumors and assessments, and to attach a variety of files such as photos, scanned documents, spreadsheets and word-processing documents. It provided powerful search and reporting capabilities.

When we created INCITE, the market for CI software was limited to spreadsheets (e.g. revenue forecasts and cost breakdowns) and unstructured document databases (which users accessed via search functions). INCITE was unique in that it provided a structured framework for competitor and market information, and accordingly, it encouraged a more structured approach to gathering, analyzing, and distributing CI within a company.

Over the years, we installed INCITE at dozens of companies, including several Ford and GM divisions and major companies in a variety of industries, particularly those reliant on scientific research (e.g. pharmaceutical companies). For an INCITE project, our consultants worked with the client's key business, marketing and technical staff to thoroughly understand the dimensions and EEIs of their business. We then customized an INCITE database to represent the identified dimensions/EEIs, and loaded sample information to provide context to the client. It typically took several weeks (sometimes months) from the start of a project until customization was complete. At that point, we installed the software at the client's site, and provided training on how to use the software and best practices for entering CI into the system.

One of our primary methods for marketing INCITE was via the Society for Competitive Intelligence Professionals (SCIP, since renamed Strategic and Competitive Intelligence Professionals). SCIP was founded in 1986, shortly before INCITE's birth. We advertised in SCIP's quarterly magazine, and we attended their annual conference, manning a booth in their exhibit space. While I had participated in sales and marketing activities for years, booth work was my trial-by-fire introduction to true selling.

Hundreds of potential customers milled about the exhibit space, and we had to cull them from the herd, and give them a short pitch in hopes of interesting them in a longer pitch. Manning a booth at a trade show teaches you to approach people more boldly, to hone your sales pitch to a few key concepts, and to shrug off the many negative responses. It is also extremely tiring work, both physically from standing there all day long, and mentally from pitching your wares all day long.

The sales process for CI products was long and complicated. For large companies, there are many stakeholders for CI, across multiple divisions, and within various departments. Often, we had to work our way through multiple divisions and departments to build a coalition of people who were prepared to make the necessary investments, and who believed in our approach. This required multiple visits and resulted in a long sell cycle - often it would take a year or more from first contact until we closed a deal. Our close rate was also rather low, because many companies could not justify the investments of time and money required to complete a CI project. In addition to a client's internal costs, the minimum fees we charged for an INCITE project was around $20,000, but for most projects, our fees exceeded $40,000.

As with most software products, we continually added features and functionality to the software. Some of our developments were to address the specific needs of a client, though when possible, we implemented any such customizations in a way that would benefit all our clients. Some developments were to add functionality that would help us sell more software, by adding features found in other products in the marketplace, or to address a wider audience.

For the Internet-enabled software we use today, software updates happen via automated processes - you open the software, and it guides you through an update process, with new software downloaded via the Internet. Some of these updates happen with almost no user intervention (e.g. Windows updates). All of our INCITE work predated the Internet, so we handled software updates differently. Once or twice a year, we issued a new software release (e.g. 'version 1.5'). We determined which features were ready for inclusion (that is, fully tested and working), and which we would defer for a future release. We developed an update program to apply any needed updates to both software and our clients' INCITE databases. And then, we produced installation diskettes.

INCITE also predated the use of CDs to deliver software (CDs were a precursor to DVDs, now also verging on obsolete). So, we delivered our new releases on 3.5" floppy diskettes, each holding 1.44MB of data (a fraction of the size of a single photo taken with a basic smart phone). As the INCITE software grew more powerful, it also grew in size. A new release might consume a half dozen diskettes. One by one, we inserted a diskette into a computer, copied a collection of files onto it, pulled it out, and put a label/sticker on it (so you could keep track of which disk was which). We created a half dozen disks for a single copy of the release, and then repeated this process to produce sets of update disks for each of our INCITE clients (eventually, dozens of them). Larger software companies like Microsoft had automated processes to produce software update disks (and ultimately, CDs), but we were too small to afford such conveniences.

Then we mailed the disks to our clients and awaited the inevitable support phone calls to address issues they encountered while attempting to install the update. Despite our best efforts, the software updates were never bulletproof - often we had to troubleshoot update and other software issues via phone. Although we did not have the Internet to connect us with our clients, we did have mechanisms to connect with them online.

Before the Internet, people connected to a variety of online services via modem and phone line. We subscribed to (paid) online services like Compuserve and AOL, and connected to them via phone numbers they provided. While today you might 'Google' a topic to find support via Internet search, before the Internet, we dialed up Compuserve, and went to a forum devoted to a topic (e.g. 'Visual Basic programming'). Software vendors created forums within these online services to handle support for their software products.

Software vendors also set up bulletin boards (BBS) for both support and file sharing. A BBS was akin to a service like Compuserve, but private and accessible only to authorized users. Quest set up a BBS to improve our support options for our clients. Whether to solve installation/update issues, or any other software issues a client might encounter, we might need to repair one of the program files, or develop a repair program specifically designed to fix a problem. Before we offered the BBS, we mailed (or sent overnight via Fedex) a floppy disk containing repairs to the client, and guided them through applying the repairs via phone. With the BBS, we placed the repair files on the BBS and clients downloaded them directly from us, speeding up the repair process.

You might ask why we didn't deliver software releases via the BBS. The answer is speed. In the 1980s, we relied upon 2400 baud modems to move data between our PCs and an online service. It wasn't until the early 1990s that we had access to affordable 9600 baud modems, and the mid-1990s for 56K modems (which remains the top speed for 'phone' modems even today). To put these numbers into perspective, my current (in 2020) Internet service offers 50MB (megabit) download speed, which is over

20,000 times faster than our speed in the 1980s. At 2400 baud, it would take more than an hour to download the contents of a single floppy disk, and that's assuming that something didn't go wrong with the download process or your phone connection, which happened with frustrating frequency. Perhaps a better example would be a typical smart phone photo which have a file size of 4MB. It would take less than 1 second to download this at 50MB, but nearly 4 hours at 2400 baud.

As modem speeds increased, we also came to offer support via remote control software like Carbon Copy and pcAnywhere. This software ran on both the client's computer and our own computers, and it allowed us to connect to the client's computer and control it, as if we were sitting in front of it (functionality now built into Windows). This was a more effective support solution, both because it was immediate and because it allowed us to work on the client's software installation directly. However, it required that the client connect their computer to a dedicated phone line/number, so we could dial in directly to that computer.

One other technology worth mentioning here is the 'software protection dongle'. Modern software can 'phone home', that is connect via Internet to an authorization process, to verify that the software is being used by the registered purchaser. Before the Internet, we used a different technique, the dongle, which was a small electronic device plugged into the back of a computer (via the printer port, in between the computer and the printer). Each time you started our software, it said 'hello' to the dongle - if it did not receive the correct response, the software shut down. So, you could copy our software and install it on another computer, but you could not run it without the dongle, and you couldn't copy the dongle (great word, isn't it?).

Wincite

Through the 1980s, all of our software development work was 'text' based - business users had not yet adopted Microsoft Windows. Everything changed in the early 1990s with the introduction of Windows 3.0, which was the first widely used version of Windows. Soon thereafter, Microsoft released Visual Basic (VB), a new programming environment for developing Windows-based software. The software marketplace was moving toward GUI (Graphical User Interface) software, so we investigated how we might rewrite INCITE as Windows/GUI software.

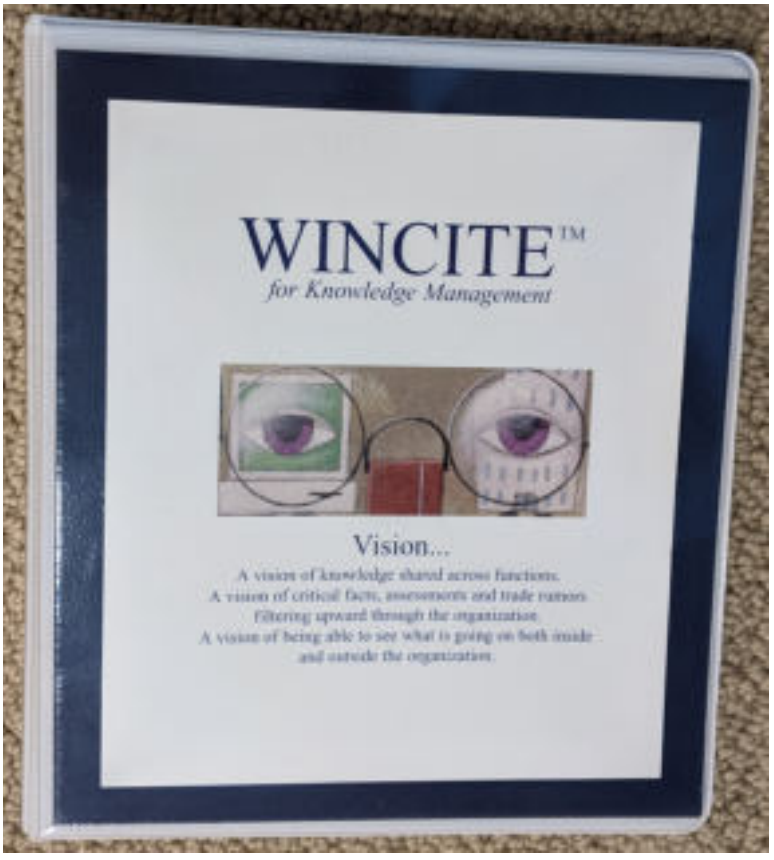
While we could keep the general look and functionality of INCITE, we had to write a new version of the software from scratch. We used a new programming language (VB) and a new database (Microsoft

Access), and we took advantage of the capabilities offered by the GUI.

Development of Wincite caused issues within our small company. We were still struggling to make enough money to pay ourselves the full salaries we were due, with me hurting more than my partners because my salary was less than theirs (due to my status as junior-most partner). Trimmer felt that we had to jump into Windows development because the marketplace would soon demand it (and he was almost certainly right). I felt that given our financial situation, we could not afford to divert resources away from revenue-producing projects.

This disagreement resulted in Trimmer assigning Windows development research to Jim Brennan, the most skilled programmer on our staff (other than me). I was working overtime doing all the work associated with our 'old' product (INCITE) while Jim got to work on the 'new' product (Wincite). This bothered me greatly because I too wanted to learn Windows development, and more so, because I wanted to get paid.

Jim produced some useful demonstrations of what we could build in Windows, but he didn't have the expertise to build the framework that we would need to complete a Windows product. As I had time, I leveraged Jim's code and his learning to jump start my own forays into VB programming. Within a year or two, we built the first version of Wincite, with most of the functionality of INCITE. Soon thereafter, all of our new installations were of Wincite, and we began to do INCITE-to-Wincite upgrades for our existing clients.



Travels

The bulk of my travels for Incite/Wincite were relatively quick visits to clients and prospective clients. We did most of the client/project work in the office, visiting clients early on to develop specifications and

later on to install the software and provide training. However, some of my travels were noteworthy.

The SCIP annual conference in 1991 was held at one of the large hotels near Central Park in New York City. We flew into JFK Airport, and noticed what seemed like more than the usual hubbub, but we thought little of it because it was New York, and we were focused on getting set up for the conference. As this was the day before the conference started, we needed only to set up our booth, and then we were free to explore the city for a few hours. It was a sunny June day, and as we wandered toward lower Manhattan, we discovered it was a momentous day. New York was hosting a ticker tape parade in honor of the veterans of Operation Desert Storm (aka The Gulf War, a 1990 campaign to repel Iraq from its invasion of Kuwait). We joined over 4 million people cheering and waving flags and enjoying the paper blizzard as the parade of soldiers and military leaders wound its way through the city.

After the parade, we had drinks at Windows on the World, the restaurant at the top of the World Trade Center (106th and 107th floors of the North Tower). The views were incredible. We headed back to the streets, which were still thriving with crowds from the parade, and clusters of partying soldiers. It was such a pleasant evening, both weather and mood, so we walked back to the hotel, along Broadway, from lower Manhattan, through Greenwich Village, all the way to Central Park. We were later told that on any other night, such a walk would be unthinkable, that suit-wearing businessmen could not travel that path unmolested. But on that night, with the streets full of soldiers and revelers, we made it safely back to the hotel.

Via our work with SCIP, we developed relationships with a few CI consultants, who became resellers of our software. The most significant of them was Andrew Pollard, a CI consultant in the United Kingdom, with contacts at some of the largest UK companies. He visited our office to learn about the software and the related consulting and installation work. I then made several trips to the UK to help him with sales calls.

Andrew was a fascinating guy. Though he didn't offer many details about his past, I got the impression that he embarked on his CI career via some version of the spy business in the UK. While most of the CI consultants we encountered were suit-and-tie-wearing business consultants, Andrew was more akin to a feet-on-the-street private investigator. On several occasions, as we were driving through England to visit a client, he detoured to the back side of a manufacturing facility to take long-lens photos of loading docks or raw materials storage.

Andrew also introduced me to 'the pub'. Trust me, I was quite familiar with good beer and the places where they serve it. But an English pub is different. The good ones are small and quaint and look like they have been there for centuries. Often there is a fire in a stone fireplace with a dog sleeping nearby. Strong-armed barmaids (and barmen) 'pull' the beer into shapely pint glasses, the tap handle pulled to draw the beer from uncarbonated kegs. The beer is warm and strong and flavorful - and while I enjoyed most of them, my favorite became 'bitter', in particular, John Smith's bitter ale.

After our last client visit of the day, Andrew would drive us home, but inevitably, we would find a nice-looking pub along the way. "Let's grab a pint," he would say, and gentleman that I am, I would agree. Men wearing suits and ties may receive wary looks when entering a small village pub, but Andrew was

so outgoing that he quickly charmed the locals. He would introduce me as his friend, "the yank", and assure them I was worthy of their pub. Andrew taught me to play pub darts, usually the game 301, in which you start with 301 points and throw darts to reduce your tally with the goal of being the first to reach exactly zero. We played 'double on', 'double off', meaning you needed to land a dart in the outermost ring, both to get started with your scoring and to finish the game. I had a knack for darts, particularly hitting a needed double, which most of the locals found entertaining, except for a few who lost to Andrew and me when playing alternate shot games.

For some of my trips, I stayed in London, and we visited London-based companies. On other trips, we visited companies throughout England, and I stayed near Andrew's home in Northampton, between London and Birmingham. He arranged for me to stay in a lovely old hotel, which backed up to pastoral fields. Weather permitting, my windows were open, and I awoke to the gentle clanging of sheep bells and bleating sheep grazing outside my window.

Though I visited England several times, I did almost no sightseeing, other than looking out the window of a train or car. However, I did visit a few memorable places, none more noteworthy than Hamleys, the oldest and largest toy store in the world. My boys were all under age six, and I wished they could be with me as I browsed the wonders throughout Hamleys seven floors. I felt compelled to buy toys for the boys, so I brought home a stuffed gorilla for Brian, a stuffed frog for David, and a stuffed tiger for Matthew. A few years later, I reveled in taking the boys there when we visited London as a family.

My most memorable client visit was to Unilever House, overlooking the Thames near Blackfriars in London, and headquarters to Unilever, a massive multinational consumer goods company. It is an intimidating building, and it offered an intimidating experience, as we met with rather stuffy and elite business people in rather pompous offices deep within the building. What was most memorable was lunch, served to us in an exquisite private dining room, one of several in the building's basement, with ornate decor and elegant table settings. Lunch was served in courses, by white-gloved servers, spooning gourmet fare from silver platters onto fine china. Lunch was taken leisurely, with small pours of wine, followed by delightful desserts. Business talk was kept to a minimum - instead we spoke about our interests, our travels, our families. Throughout my career, I have never had a similar experience.

Another memorable trip was to Italy, to install our software at a subsidiary of Whirlpool Corporation. I flew into Milan and then took a taxi to Comerio, where Whirlpool had a plant near Lago di Varesse, an area far more beautiful than befits an appliance factory. The taxi ride was the most harrowing 40 minutes of my life as the driver ignored signage and lane lines, sometimes hurtling down the center line of a two-lane road to pass one car while another whizzed by in the other direction.

Having flown from the U.S. to arrive in the afternoon, I was tired and I asked my hosts if we could dine early so I could recover before starting work the next morning. They said we might find a table at 7PM, but not earlier. I freshened up at the hotel, and they took me to one of the excellent nearby restaurants. Ours was the only car in the parking lot. A group of people were eating at a picnic table outside the restaurant. My hosts asked if we might be served, and after some gesturing and negotiating, the people led us inside. It turns out that they were the restaurant staff, having their dinner before opening the restaurant. Dining in Italy doesn't truly begin until 8 or 9PM. Despite our earliness, they served us a brilliant meal, the highlight of which was a risotto infused with wild blueberries. It was not until we were nearly done with our meal that other customers entered the restaurant.

The next morning, I had breakfast and arrived at the plant to begin work. After introductions and some preliminary work, the client staff asked if I wanted some coffee, to which I replied, "oh, absolutely." I was imagining a big cup of coffee to sip while I worked, but that's not how coffee works in Italy. Instead, we walked to the main entry to the plant where, behind the reception area, they had a coffee bar, complete with a fancy espresso machine and barista. We enjoyed freshly made espressos, and they allowed my unorthodox request for a second cup, which I needed to help cure some jet lag.

Also noteworthy was travel I did not do. We sold our software to Uniban, one of the largest suppliers of bananas in the world, with headquarters in Medellin, Columbia. Yes, Medellin, also home to Columbia's notorious drug cartels. As a family man with two young children, I could not in good conscience even consider going to Medellin. Although it seemed a bit out of character for him, our only volunteer was our programmer, Jim, who made a couple of trips there. He said it was a beautiful city, and that Uniban was a very professional company. I have no regrets about skipping that one.

I'll finish my travelogue with a trip I made to Brussels, Belgium, for a conference. My family lived in Brussels for two years in the early 1970s, and this was my first trip back. I reveled in seeing the sights and revisiting familiar places. I enjoyed the sounds of trams squeaking on the rails and people speaking French and Flemish. I drooled at the smell of the gaufre (a true Belgian waffle) wafting from a street-side stand (yes, I ate one, and it was warm and sweet and delicious). I wanted to revisit all of my favorite Brussels experiences, and then to share stories with my family upon my return home.

Topmost on my list was a visit to Chez Richard, one of several small seafood restaurants in the Îlot Sacré, a touristy area near Brussels' iconic Grand Place. My family went to Chez Richard regularly while we lived there, and we almost always ordered moules provençale, mussels served in a broth of tomatoes and herbs, and served with Belgium's famous frites (french fries). I meandered my way from my hotel, through the Grand Place, to Chez Richard, where my seat gave me a view of the entire restaurant and out the front window. I ordered a Stella Artois (a local Belgian beer and family favorite), and moule frites provençale. While I waited for my meal, I sipped my beer and reminisced.

The waiter delivered a good-size pot and a plate of frites, and he lifted the pot's lid to reveal a beautiful mound of mussels in broth. As is my tradition, I placed my face directly over the pot, the steam dampening my face as I took in the heavenly scent. And then I cried, tears trailing down my face, as I sat there, more lonely than I had ever been. While I enjoyed the food and the remembrances, it was heartbreaking to experience Chez Richard without my family there to enjoy it with me.

Offices and Staff

Quest's first office was in downtown Birmingham, Michigan, a space we sublet from a financial firm that had moved elsewhere. We had roomy offices for each of the partners, plus extra offices and a large open space that was previously used for cubicles. For us, this space was useful for games like soccer and bowling when we needed a break from work. We got a sweetheart deal on some quality (but old) furniture the financial firm left behind, and just like that, we were in business.

Lisa and I bought our first house, in Harper Woods, on the east side of metro Detroit. My commute to Birmingham was along 8 Mile Road (yes, the 8 Mile made famous by Eminem), a busy 8 lane road, in constant need of repair, dotted with stoplights along its length. Highway 696 was under construction a couple of miles to the north, and when they completed its first section, it shortened my commute time considerably.

Our office sublet was only for a few months, so we moved to permanent office space in nearby Troy. We found space in a new building, a green glass 10-story building, on Big Beaver Road, just east of Somerset Mall. Because it was a new building, we had the luxury of designing our own office space, which the building owner built out to our specifications. Given our ambitious business plans, we had plenty of space for our three partners, and room enough for up to another 10 staff. The next building over housed a gym, where we played racquetball, both to stay (get) in shape and to take breaks from our demanding work. The gym also offered a pool, and I enjoyed getting back into swimming (not having done any significant swimming since high school).

When our three-year lease expired, and with our ambitions tamed by reality, we moved to more affordable (and smaller) space a few blocks away, on Kirts Blvd. We remained there until we sold Quest and moved the company to Chicago. While it served its purpose, this was not my favorite office, as my view was of a parking lot and the I-75 highway beyond.

Over the years, we hired a few programmers, consultants and salespeople. Early on, we hired a few programmers to help build our EIS software, but as our EIS ambitions faded, most of them moved on. As our business shifted to consulting, and then to CI, we hired a few consultants. And throughout Quest's existence, we hired, and then fired, a steady stream of salespeople. With great optimism, we hired a salesperson, brought them up to speed on our product and service offerings, and sent them out to find us new clients. One after another, it became evident that they were incapable of understanding our offerings well enough to sell them. The only way we could sell anything was to do it ourselves.

On to Greater Things - Again

Trimmer, Jenks, and I started Quest in mid-1987. In 1991, Jenks decided to move on from our startup software company, to earn a consistent (and larger) paycheck at IBM. I can't blame him, and there's a part of me that wonders why I didn't do something similar. But Trimmer and I saw hope in the CI business, so we continued on. However, by 1993, our business was doing little better than surviving. Trimmer began looking into other opportunities, including another CI-oriented product called Grapevine.

At first, he attempted to build a Grapevine practice within Quest, but I must admit that I stonewalled him, as this was a major distraction from our work with INCITE and Wincite.

Eventually, Trimmer tired of our continual struggle to make Quest successful (profitable), and he probably tired of wrangling with me on how to do it. In late summer 1993, Tom announced that he was leaving Quest to take a position at Ernst & Young, another of the Big 8 accounting firms. He said that I should continue running Quest, and that he would provide as much support as he could. I took the situation differently. I felt that he had abandoned me and the company, and I didn't want him involved (meddling) with it anymore.

Our partnership agreement included a clause that said if you no longer work full time for Quest, you "shall be deemed to have offered your shares for sale to the other partners." I told Tom that I was exercising this clause, and that we should negotiate a buyout. Tom was not happy to hear this, because he wanted to keep his 2/3 ownership of what would be a side business for him. He knew a buyout would likely be an unpleasant process, and that he would receive minimal compensation. Tom set out to find a better offer.

We met with Craig Bickley and Sam Mancuso, who were partners in a few local technology businesses, and whom I had met through my father, who bought computer systems from them for his public relations business (Publicom). We'll revisit Craig and Sam later on (see Marketplace Solutions), but they passed on investing in Quest. Tom made a few calls to his contacts and found one company that might be interested in buying Quest: Braun Technology Group (BTG).

BTG was primarily owned by Steve Braun, a former PW consultant, who built BTG into a substantial consulting firm, ultimately taking it public before selling it to another larger company. Braun was advised by Don Smith, who had been Steve's boss at PW (before Don retired from PW). Don and Tom had been partners at PW, Don in Chicago and Tom in Detroit. They had collaborated on several projects, and knew each other well. Don and Steve decided Quest would be a good fit with BTG, so we negotiated a deal, both for BTG to buy Quest, and for BTG to hire me as an employee. This was a boon for me, because not only did I finally receive a steady paycheck, I also received my share of the proceeds of the sale, which included both a lump sum payment when we signed the deal, but also quarterly royalty payments based on Quest's sales.

Career - 4. BTG / Wincite

We completed the sale of Quest to BTG in November 1993, and BTG renamed the company Wincite Systems, because selling Wincite software and services had become our sole product offering. In early December, I started to work part time at BTG's office in the Inland Steel Building in downtown Chicago, at the corner of Monroe and Dearborn, across from First National Plaza (famous for the stairway dance scene from the movie Ferris Bueller's Day Off). BTG's offices took up most of the 3rd floor, and they provided me an office with floor-to-ceiling windows overlooking the plaza. While we couldn't see the expanse of downtown Chicago from the 3rd floor, we had great closeup views of everything happening on the street below us.

After we signed the paperwork, I became all things Quest. When Trimmer left Quest, I took over his roles, in addition to my own. I was to be the only Quest employee moving to Chicago, although our programmer (Jim) agreed to stay on for a few months to assist with the transition. For a couple of months after the sale, we continued work as usual in Quest's Troy Michigan office.

Moving to Chicago

In February 1994, we moved the Quest office to Chicago, scheduled over a weekend, so that our clients would not notice any interruption in our services. I rented a large U-Haul, and my remaining staff and I loaded Quest into the truck. My load included computers and printers (both of which were larger back then), a large copier, many boxes of books and documents, and some furniture (the rest of which we had sold). As we neared the finish, it began to snow, and soon it was snowing heavily. On a Friday afternoon, right at the start of rush hour, we closed up the truck, and I headed for home. In less than an hour of snowfall, the highway was already thick with snow, and traffic crawled. It took me over two hours to get home (a drive that normally took half an hour).

The next morning, we drove the truck to Chicago, with Lisa's dad Herb riding with me in the truck, and Lisa ahead of us in the car. Fortunately, the snow had stopped falling overnight, and the highways were clear. The same could not be said for downtown Chicago. When there is major snowfall in a big city, they can't just plow the roads, as the snow/slop would remain on the sidewalks. For a heavy snowfall like that one, they close the city streets, one at a time, and use front loaders and dump trucks to haul the snow away.

Dodging a few closed streets, we worked our way from the highway to my new office building. Through 6+ inches of snow, I eased the truck into the alley behind the building, and to the loading dock for the building. We got some help from BTG staff, and we had the truck unloaded within an hour. Then it was time to get the truck back out of the alley. Large moving trucks are rear-wheel drive. When fully loaded, they get good traction in the snow. When empty, they do not. I had to back the truck down the alley, across the sidewalk, and onto Dearborn Street, a 3-lane, one-way street. The truck did OK in the alley, but spun its wheels at the sidewalk, which held a pile of snow awaiting the front loaders. Eventually,

Herb stood in the middle of Dearborn to hold up traffic, while I gunned the truck in reverse to lunge across the snow and into the street. It seems that every move entails some form of adventure.

For the next few weeks, until Lisa and I completed our move to the Chicago area, Jim and I commuted from Detroit on a weekly basis, Monday to Friday in Chicago, and weekends back in Detroit. In Chicago, Jim and I stayed at the Chicago Athletic Association (CAA), of which Steve Braun was a member. The CAA was around the corner from BTG's office, on Michigan Avenue, facing Grant Park and Lake Michigan beyond. The CAA building was historic and architecturally significant, and it's membership included many of Chicago's elite business people. While it was recently (2015) transformed into a high-end hotel, when we stayed there, it offered only a few rooms for overnight guests, and these rooms were drab and dated. But it worked well for us and was a fraction the cost of nearby hotel rooms. One bonus of the CAA was that we enjoyed many dinners there, in an elegant dining room little changed in half a century.

Lisa made a few trips to Chicago to lead our search for a house. As in most major cities, there's an inverse relationship between commute time and affordability - the shorter your commute time, the more expensive the housing. We settled on a nice Tudor in suburban Wheaton, in a neighborhood recently carved out of the adjacent cornfields, and much less expensive than the walk-to-train homes in downtown Wheaton. Given the logistics of buying in Wheaton and selling in Harper Woods, we weren't able to move the family to Wheaton until April 1994.

With no traffic, it was a 45 minute drive from Wheaton to downtown Chicago. But during rush hour (or more accurately, rush **hours**), it was a minimum hour and half drive, and weather or accidents often led to a drive of over 2 hours. Another problem with driving was parking downtown. Again, there was an inverse relationship between proximity to office and affordability - the closer to the office, the more expensive the parking. To save a few dollars a day in parking costs, you might have to walk 10 minutes or more from the parking lot to the office, which was most unpleasant during Chicago's brutal (windy!) winters.

Instead, I opted to commute by train. I left the house each morning at 6:45AM, drove to a parking garage near Wheaton's downtown train station, and walked from the garage to the station to await my train. The express trains completed the trip downtown in about 40 minutes, disembarking passengers at Northwestern Station (since renamed OTC). I then walked 15 minutes from the station to my office, often picking up a Starbucks coffee on the way (Starbucks was a relatively new concept back then). Door to door, the trip took an hour and twenty minutes. In the afternoon, I reversed the process, and strove to catch the 5:10 express train so that I could get home by 6:30PM. If I missed that train, I had to wait 10 minutes to catch a slower train (due to more stops along the way), and I wouldn't get home until after 7PM. I'll confess that after a particularly challenging day at the office, I might buy a can of beer from a bar in the train station as a rare treat for the ride home.

Every day, I carried a full briefcase and a luggable computer (20+ pounds) back and forth to work. I often worked at home after the kids went to bed, so I needed my computer with me both at work and at home. I couldn't do computer work on the train, because the computer was too big and bulky, and because it had no battery - it had to be plugged in to an outlet. Instead, I did paperwork or read

computer magazines - we didn't yet have smartphones, so I did my reading from paper magazines. Yes, I hauled 30+ pounds of stuff between my office and the train station (15-minute walk) twice a day, five days a week, and as you can imagine, I have since suffered shoulder and back issues.

Jim continued his weekly Detroit to Chicago commute for a few months. At the end of his stint in Chicago, I took him out to dinner, as a well-earned thank you for his help throughout our move to Chicago and BTG. I took him to Ruth's Chris Steak House and told him, "no limits". We enjoyed a 48 ounce steak for two, carved table-side, and served in a pool of herbed butter. It was a worthy sendoff. While researching this article, I found out that Jim died in July 2020 of leukemia, at age 68. He was a good human and a talented programmer. Jim also pursued an unusual hobby: cataloging all of Michigan's official historical markers (see www.michmarkers.com). He will be missed.

Transition

For my first several months at BTG, I was the only one who had a clue about what we were doing. I was the primary contact for our existing customers. I did the consulting work to customize our installations. With Jim's help, I provided technical support and did software development. And after Trimmer left Quest, I also handled sales and marketing. In retrospect, I can't conceive how I did it all. After we moved the office to Chicago, we hired staff, and eventually, this reduced the burden on me.

BTG's principals, CEO Steve Braun and his mentor/advisor Don Smith, planned to integrate Wincite Systems into BTG's existing consulting business, with resources shared between BTG and Wincite. Ultimately, we had minimal sharing of resources, but Wincite drew a couple of staff members from BTG's ranks. First, we took on a salesperson who proved utterly incapable of understanding Wincite's software or consulting services. After a month or two of awkward sales meetings, BTG let him go. It turns out that he hadn't been much help in selling BTG's consulting services either. Next, we took on one of BTG's consultants, and he proved much more effective, helping me with client installations and support. Gradually, we hired a few additional consultants and software developers, and in late 1994, we hired another salesperson, Tim Taschler.

During my years with Quest and Wincite, we hired a succession of salespeople, each of whom came aboard with great bravado. I suppose that's the essence of salesmanship, that you must be bold and believe in your product, particularly when the product is yourself. However, none of the salespeople we hired proved capable of selling our wares, and we fired all of them after a few months. Part of the problem is that it takes a consultant to sell consulting services - you need to discern a prospective client's problems, AND offer a workable solution. The salespeople we hired did not have the business and technical acumen needed for the job.

Another part of the problem is endemic to salespeople everywhere: selling what the prospect asks for instead of the solutions we offer. In that our salespeople were not consultants, they could not guide prospects to see our offerings as a solution to their problems. Instead, they would return from a sales meeting and say something like, "They'll buy our software if it can do 'X'. We can do that, right?" Invariably, doing 'X' required extensive software development work that would take time and resources we could not spare, and for which the prospect would not wait or pay.

Tim was different. He quickly learned the capabilities of our software and the spectrum of solutions we could provide. At first, I was not happy with Tim or his role as VP Sales. Don Smith hired Tim without my involvement, and he brought Tim up to speed without my input because I was busy doing Wincite's work. The two of them took on our precious sales leads, taking over the role of sales from me, and they set into changing our marketing materials and sales process. I never liked Don's participation in sales calls, because he wanted to lead the discussion despite knowing little about CI consulting or the Wincite software. I worried that Don and Tim would ruin our sales leads.

Fortunately, Tim quickly realized Don's shortcomings, and worked around him. It took a few months, but I came to consider Tim the most gifted salesperson I have ever worked with. He understood the gamesmanship that is sales, and he shared many insights with me. His most important lesson for me was about pricing. As consultants, we set hourly rates for our services, and then charge clients based on the hours required to complete the work. Tim offered that pricing should be a function of value to the client, not just the cost of a consultant's time. He said that an arbitrarily high price tag was no different than an arbitrarily set billing rate - the price a client pays does not have to be tethered to the hours needed to provide the solution. He had the audacity to ask prospects how much they had budgeted for a project, and with startling regularity, they told him. As the budget was almost always higher than what we needed to make a profit, Tim told them he could provide the solution for under the budgeted amount, thereby making them heroes for not spending their entire budget.

With Tim in charge, we made more sales, with much higher prices - we prospered.

The Ugly Ending

Over time, the relationship between Wincite and BTG grew awkward. BTG had a high-growth strategy that would lead to them going public a few years later. While successful on its own, Wincite didn't fit in with BTG's business - their clients didn't need Wincite and our clients didn't need BTG's services. Steve appeared to lose interest in Wincite, and he focused his attention almost completely on BTG, while encouraging Don to take over the management of Wincite.

Despite our management struggles, Wincite grew quickly, both in sales and staff. Tim and I were busy traveling the country, so we could mostly avoid Don's meddling. But Don felt he was losing control of Wincite to Tim and me, and a rift grew between us. Tim and I sought a Plan B that would not include Steve or Don. Though this sounds somewhat mutinous, we had Steve's approval throughout.

We found a company (TelTech) whose visions more closely matched ours, and we procured an offer from them to buy Wincite, for more than double what BTG had paid less than 2 years earlier. In theory, the sale would have served BTG well - to recoup their investment while getting rid of a problem child. But Don had Steve's ear, and for reasons I've never been able to discern, Steve turned down the TelTech offer in favor of his deal with Don. I never learned the details of that deal, but I later discovered it was for less than half what TelTech offered.

During the TelTech negotiations, the situation at Wincite became unbelievably ugly. The other Wincite staff aligned themselves with Don - a likely result because Tim and I were so often out of the office, and

the suggestion that we were trying to take over the company and leave them behind.

This debacle unfolded during the summer of 1995. I handled the preliminary negotiations while standing at a pay phone at Silver Lake Sand Dunes, where my family was enjoying a vacation with a few friends and their families (all of us with children under age 8). While the kids frolicked at the beach, I worked the phone, having back-and-forth conversations with Tim and Steve and our contacts at TelTech.

A few weeks later, Lisa and I took the kids on vacation at the Jersey Shore, along with her brother Eric and his family. Again, while the kids frolicked at the beach, I worked the phone, this time discovering that the TelTech deal was doomed and that Don was taking over the company. While I was still in New Jersey, Tim called to let me know he had been terminated. As I was still the lead consultant and the lead software developer, I thought they would not dare terminate me - or would they?

I returned from vacation, and with great apprehension, I went into the office late Monday morning. Something was clearly amiss. The staff avoided me, and those I encountered were standoffish. Soon after I arrived, Steve summoned me into his office. He told me that Don was taking control of Wincite, and that because Don and I could not get along, I would have to go. He apologized for the situation, and then he fired me. I packed my personal items in a box, and turned in my keys, and received an escort from the building. I had called Lisa to come pick me up, because I couldn't carry everything home via my usual train commute. We drove home in quiet conversation, dismayed at the events of the day, and concerned for what may lie ahead. It was a completely clean break - I never did another thing with the Wincite software or with any of Wincite's clients.

A few years later (1999), I discovered that Don had taken on an investment partner, and they had purchased complete ownership of Wincite from BTG. My employment and termination agreements with BTG included a 'participation' clause that provided if BTG were to sell Wincite, I would receive a portion of the proceeds. I hired a lawyer, and we threatened a lawsuit to go after my share. It took nearly a year of legal wrangling to get them to agree to a settlement, but in the end, I received a sizable payment. Trimmer and I had already struggled with BTG and Wincite over royalty payments they owed us via BTG's purchase of Quest. We threatened legal action for that as well, and ultimately received nearly all that we were due. I'll confess that I took more satisfaction in beating Don than in depositing the checks.

While researching this article, I found out that Steve died in 2022 and Don in 2023. I have nothing more to say about that.

Travels

My sales and installation work required extensive travel, sometimes multiple trips per week. For example, here's my travel schedule for April 1994:

4/3 - Returned from family vacation in Florida
4/4 - Providence, Rhode Island
4/7 - Minneapolis, Minnesota

4/11-13 - New York City
4/14-17 - Moved into our new house in Wheaton
4/19 - Akron, Ohio
4/21-22 - New Jersey
4/26-29 - Boston, Massachusetts

This was typical of my travel throughout my time at BTG. When possible, I stacked multiple meetings into a single trip, so I might visit an existing client in the morning, and then drive across town for a sales call (or two) in the afternoon. Yes, I racked up many frequent flyer miles, and I excelled at getting through airports and into rental cars, and I found my way to office buildings across the country - without benefit of GPS or Google Maps.

One of my most frequent destinations was New Jersey, because it is home to dozens of major pharmaceutical and chemical companies, which were among our largest target markets (due to their extensive research operations). Fortuitously, Lisa's brother Eric lived in New Jersey, and for a few of my visits, I stayed with him. Staying with Eric was homier than a hotel, and I enjoyed visiting a few of Eric's favorite bars and restaurants. Most of my other trips were rather lonely, with me dining in a hotel bar/restaurant, and then working late into the evening.

Career - 5. Marketplace Solutions / Marketplace Commerce

My sudden and unexpected exit from Wincite Systems left me unemployed and adrift. For 10 years, I had been consumed in following a career path, with two years in apprenticeship at Price Waterhouse, and then eight years in entrepreneurship at Quest and BTG/Wincite. Every single day, I had software to develop, customers to support, and a business to build. The day Wincite let me go, I stepped into a void - there was no software, no customers, no business - just me, with no plan for what to do next.

While we had enough savings to cover a few months with no paycheck, I needed to get a job. Lisa was still on hiatus from her teaching career, serving as stay-at-home Mom for our sons aged 7, 5 and 3. She earned some money via part-time tutoring, but we were almost completely reliant on my paycheck. That first day, Lisa and I dealt with the shock of my firing, worrying about the future, but assuring ourselves that everything would be OK. The next morning, I got to work.

Job Search

In 1995, looking for a job was much different than it is today. We had online services (e.g. AOL), and the nascent Internet, but very few job listings appeared online. Instead, we had employment agencies and newspapers. Companies still use employment agencies to find and vet employment candidates, though they now have much better tools at their disposal (e.g. LinkedIn, Indeed, etc.). But newspapers, which used to reign over the job search marketplace, no longer serve any significant role.

Every day, and especially on Sundays, newspapers were filled with dozens or even hundreds of job listings in the classified ads section of the paper. Classified ads are text-only advertisements, arranged by classifications like 'Used Cars', 'Real Estate' and 'Employment', and crammed into narrow columns with tiny fonts, often filling several newspaper pages. Most classified ads have since been replaced by various online methods, but in 1995, classified ads were the primary method of connecting employers and job hunters.

Job hunters like me pored through the densely packed ads, circling any job listings worth pursuing. Because newspapers charged for ads based on word count, most of the ads were terse. As word count went up, so did the cost, but often so did the quality/pay of the job. Most of the listings included very little information about the company, and only enough information about the job to entice the right caliber of candidate to apply. As the Internet was still in its infancy, few companies had websites, so it was often difficult to find out much about them.

In 1995, email was just coming into common use, so it did not play a role in my job search process. Some job listings offered a phone number, and many offered a mailing address, but in either case, to pursue an opportunity, I mailed a resume and cover letter. While today a job seeker maintains a vigil

over their inbox hoping for a response, my vigil was over home 'land line' phone and answering machine.

I found several intriguing job listings, and I mailed them my resume, and I waited impatiently for responses. After two weeks, I had my first job interview, and over the next few weeks, I had several more. As with any job search, some opportunities were unappealing, but a few seemed promising. I discovered shortcomings in my programming knowledge, because I had spent years writing software in what were considered non-standard programming environments. But I soon realized there were worthy software development jobs in the Chicago area, and that I would be a good fit for many of them.

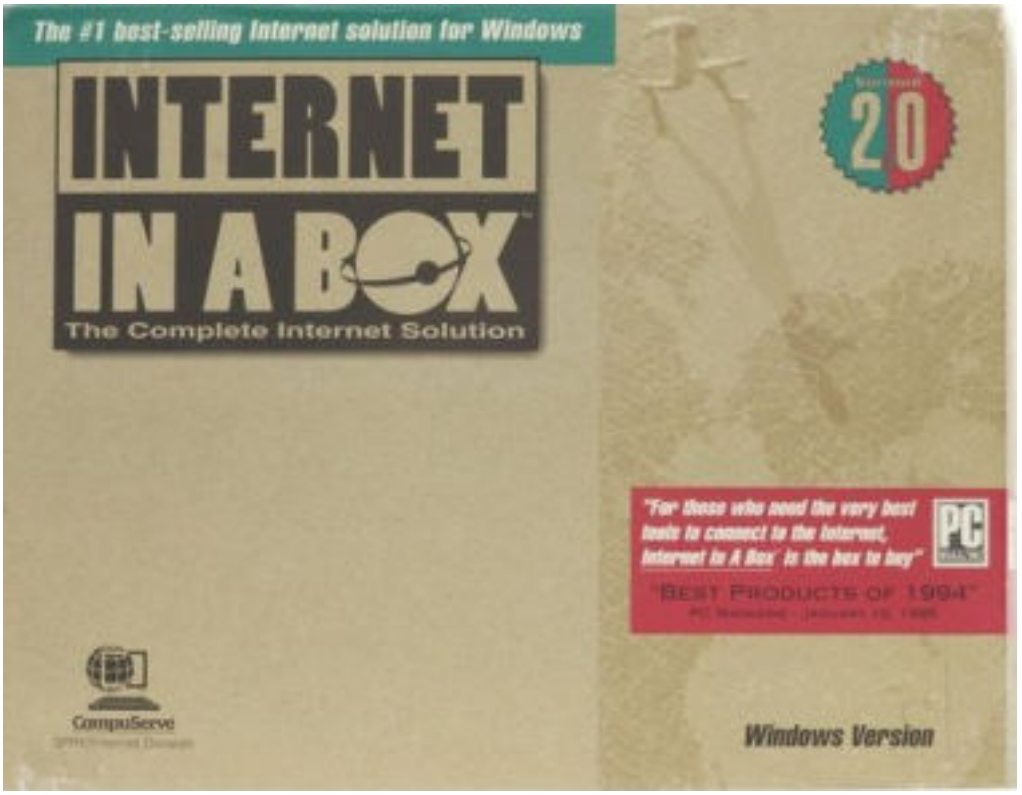
I also discovered that I was not spiritually prepared to work as a lesser cog in the big-wheel companies where I was interviewing. I had experienced this phenomenon once before, when Quest was struggling financially, and I was not getting regular paychecks, and I interviewed for a couple of jobs. During both interviews, my interviewers discerned that I wasn't truly ready to leave Quest to take on the challenges of a new job. Back then, my heart was not in my job search, but I did still have a job. Having become newly jobless, my heart needed to be in the job search, yet still, I was reluctant.

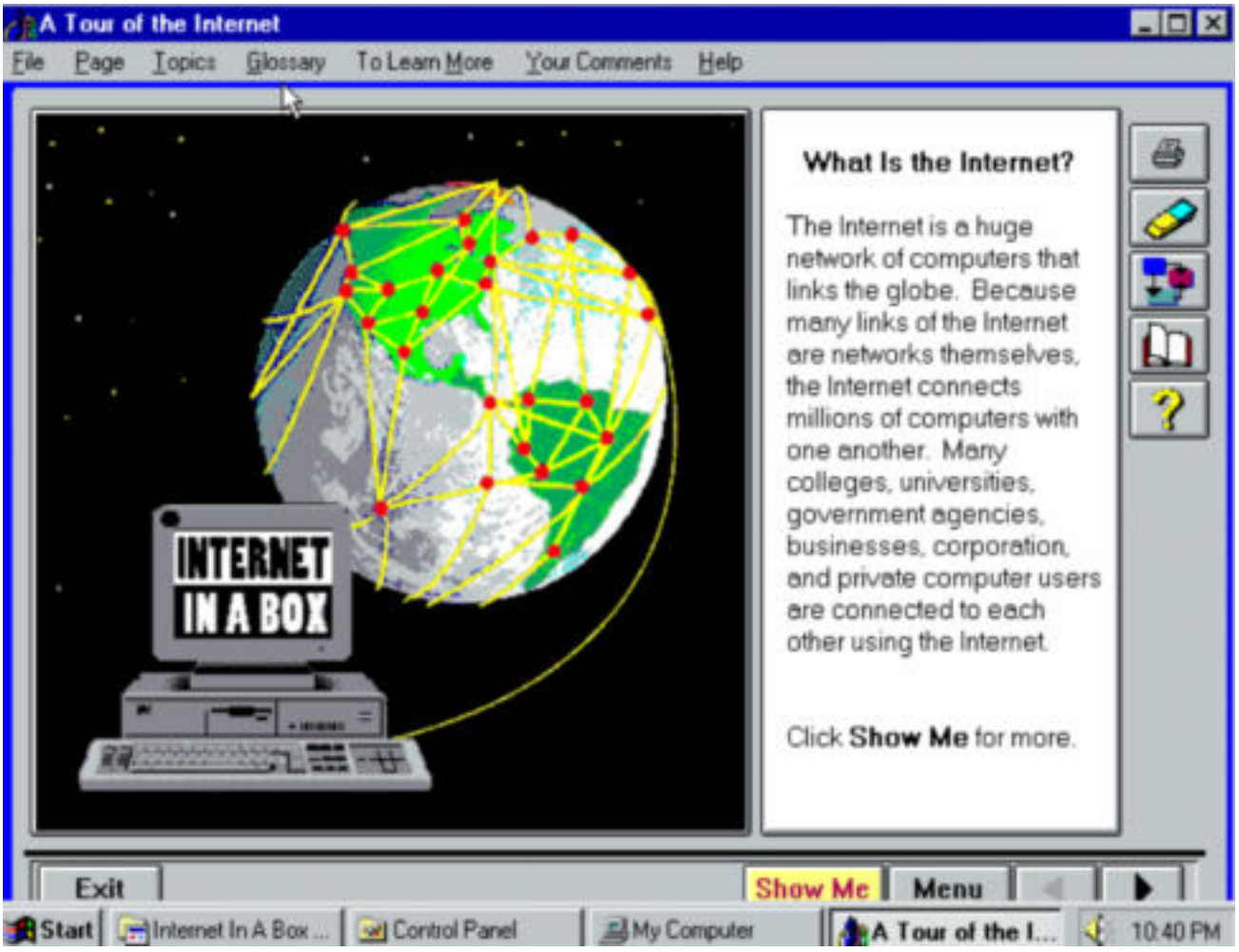
I have never been content to execute someone else's vision of what to build and how to build it, but rather I need to participate in the process. Even as a rookie at Price Waterhouse, I was encouraged to fully participate in the software development process. From when I started writing pizza store software in college through to retirement, I have always driven my own bus. If I had a different temperament, perhaps I might have taken part in great software developments at other companies, and risen in rank to manage software development teams, and reaped great financial rewards, but that was not my way. I have no regrets, as I was true to myself, and I provided well for my family (most of the time).

Despite these misgivings, I interviewed often and well, and I received a couple of job offers. But throughout my job search, I was also working on a Plan B.

Discovering the Internet

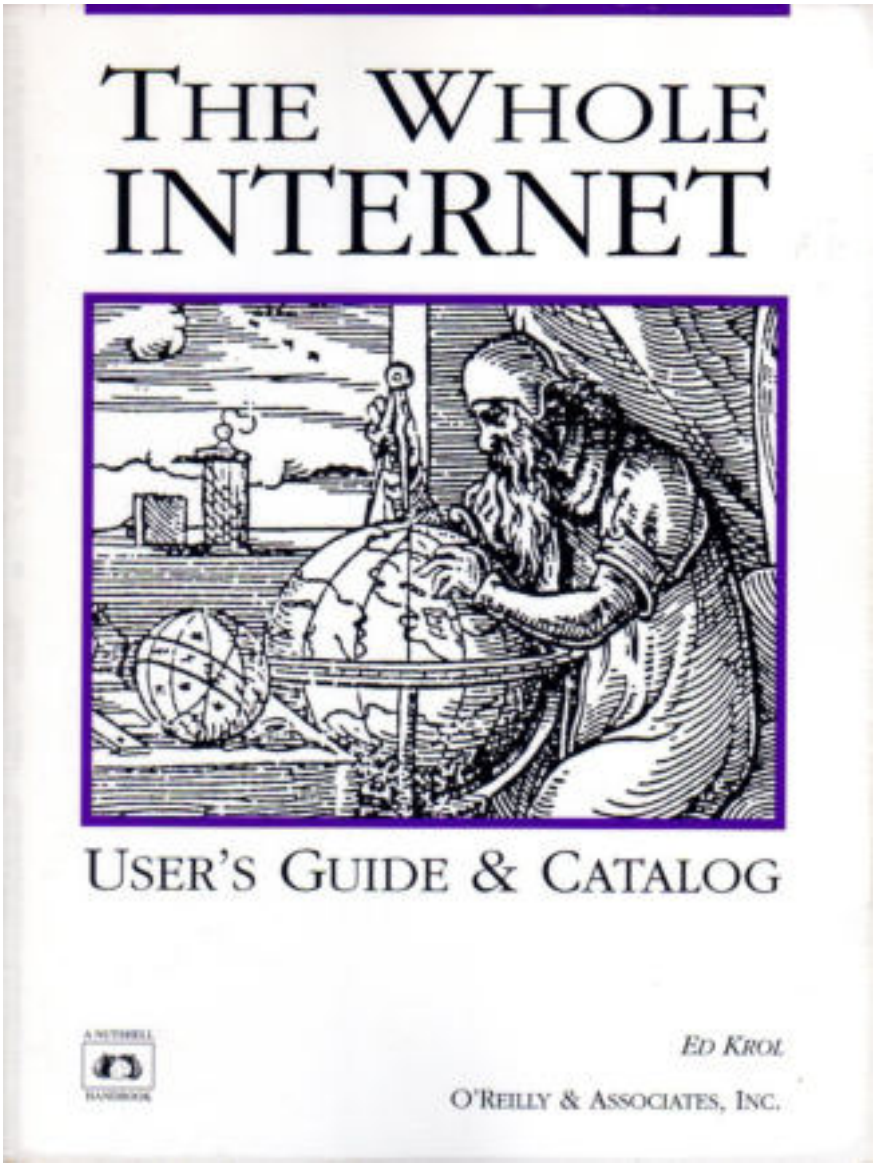
While looking for a job, I used some of my extensive spare time (a luxury unknown to me for years) to explore a new technology phenomenon: the Internet. I had been 'online' for years, via Compuserve, a paid subscription service, which offered web-like content (company pages, forums, etc.), but within a closed/private environment. I heard about this new 'Internet' (a.k.a. 'World Wide Web'), which it was said, would replace everything 'online'. To be honest, Luddite that I am, I had resisted exploring the Internet, a tool with unknown usefulness, in favor of continuing to use my tried-and-true Compuserve. Newly freed from work obligations, I now had time to learn a few new things. I went to my local bookstore (very popular places back then), and purchased a product called 'Internet in a Box' (IBox).





IBox included all the software you needed to connect to the Internet and take advantage of all that it offered, which in 1995, was a tiny fraction of what the Internet offers today. The Internet was so new that the box came with a 500+ page manual, not just for the software, but for the Internet itself: The Whole Internet User's Guide & Catalog. The book served as a useful guide, because in 1995 we did not yet have search engines to help us find content (Google didn't launch until 1998). Instead, we found

content via print media (books, magazines and newspapers) and human-curated directories of websites - something like Netflix categories and recommendations.



The software provided by IBox included 'Mosaic', the web browser software that precipitated the explosive growth of the World Wide Web. Mosaic later morphed into Netscape Navigator, which for a few years was so dominant as to be almost synonymous with 'Internet'. Also included was a rudimentary email program and a 'news reader', which provided access to discussion forums (a.k.a. 'Usenet') that proliferated across the early Internet. IBox required that you already had a modem (minimum speed 9600 baud) connected to a phone line, and that you paid a monthly subscription for dial-up connection.

During my preliminary studies of the Internet, I didn't realize that it would soon become my profession.

Plan B - Knowledge Systems, Inc.

I admit that I held a grudge against Wincite Systems, and that I craved an opportunity to beat them at their own game. At Wincite, I was constrained within the capabilities and direction of our existing software. We needed to support the requirements of our existing and prospective customers, and this

precluded making profound changes to the software. On my own, I had the freedom to envision a major leap forward, to address capabilities far beyond Wincite's.

When we realized our days at Wincite were numbered, my compatriot Tim Taschler and I collaborated to develop a business plan for Knowledge Systems, Inc. (KSI), a new company which would create and sell knowledge management software and services. I created software development plans and Tim created sales and marketing plans. We pitched our ideas to TelTech, but they had been burned by BTG's duplicitous negotiation strategies. TelTech attributed some of the turmoil to Tim and me, and they declined our proposals. While I think they missed out on a great opportunity, I don't blame them.

Tim and I explored our networks for opportunities to fund our plans. I contacted Craig Bickley and Sam Mancuso, who were partners in a few Detroit-area technology businesses, and who had considered purchasing Quest Management Systems before we sold it to BTG. I had known Craig since the early 1980s when his company sold computer systems to my father's public relations business (Publicom), and I had served as my father's technical advisor.

Although Lisa and I had been in Chicagoland for less than two years, we wanted to move back to Michigan, to be closer to family. We had grown tired of missing family gatherings and birthday celebrations because we were too far away for a quick visit. If I could persuade Craig and Sam to invest in KSI, then I could accomplish both goals: leveraging my expertise in knowledge management (and yes, taking on Wincite) and returning my family home to Michigan.

I met with Craig and Sam, and they showed interest in the KSI concept, as well as the possibilities offered by the Internet. Given my employment situation, I had to give them an ultimatum - either they found a way to hire me, or I would have to accept a job offer I had in Chicago. In less than two weeks, we hashed out a plan for a company, negotiated agreements, and I had a new job back in Michigan.

Sam was a sales executive at Sun Microsystems, a pioneer in offering the hardware and software powering the early Internet. He and Craig owned Marketplace Solutions, which provided a full suite of hardware and software (Micros brand) for the hospitality industry (e.g. restaurants, hotels, casinos, stadiums). Craig managed the company and Sam helped with strategy and investment. They hired me as an employee of Marketplace Solutions, with the promise that we would soon form a separate software company in which I would have an ownership interest. It would take almost two years, but eventually, they kept their promise.

My termination agreement with Wincite included a non-compete clause, which expired a year later, at the end of 1996. Because of this, Craig and Sam would not commit to pursuing the KSI concept as our first project, which meant that they could not offer employment to Tim. Needing to regain a paycheck, Tim soon accepted a sales position elsewhere. We never worked together again, although we collaborated on a few possibilities, and I did website work for a couple of companies where he worked.

After Marketplace hired me, we explored a few avenues of software development that we might undertake. One idea was to develop software that we might sell to or through Sun Microsystems. In particular, we considered development of software in the same genre as Lotus Notes, which was

becoming dominant in the world of knowledge management, to give Sun a product offering to compete with Lotus Notes. However, all the Sun hardware and software was based on Unix, but my expertise was in Windows software, so this idea was short-lived.

Another short-lived idea was to develop Windows-based Point-of-Sale (POS) software, to replace the character-based software that dominated the hospitality marketplace. I did some experimentation with touchscreen monitors and Graphical User Interface (GUI) software development, but we soon abandoned these efforts to focus on the explosive growth of the Internet.

Moving to Metro Detroit

Much like our move to Chicago, the new job preceded our move by a few months, so I commuted. I started working in Farmington Hills in December 1995, but we didn't buy a house until February and we couldn't move in until April. We sold our Wheaton house in late March, and were fortunate for the opportunity to move the family into my sister's basement for the two-week gap between moving out of the Wheaton house and into the Farmington Hills house (thanks again Betsy & Joe!). Between December and April, I flew to Detroit for the work week and back to Chicago for the weekends, leaving my car in Detroit for transportation there, and relying on limo service to get to and from the airport in Chicago. I should also thank my wife Lisa for managing the family and household for months at a time during these transitions - wrangling three small boys while coordinating an interstate move was daunting, but she handled it all brilliantly, twice.

After surviving Chicago's hour and twenty minute (each way) commute for a couple of years, Farmington Hills was a welcome change. My new commute was 8 minutes each way, with no trains, no parking garages, no traffic. My office was less than 10 minutes away from the kids' schools, so I could attend all of their activities, even those during the middle of a workday. We immersed our family in the Farmington community, joining the neighborhood swim/tennis club, a variety of sports/teams, and the local Unitarian Universalist church. Ultimately, we lived there for 15 years, until after all three of our sons graduated high school.

Internet E-Commerce

At Marketplace, we quickly realized our best opportunity for success was via the Internet, and in particular via e-commerce, which was at the start of its gold-rush days. 1996 was the first full year in operation for both Amazon and eBay, and at that time, Amazon sold only books. Craig and Sam and I believed many companies would want to sell products via the Internet, and they would need e-commerce software to do the job. So we embarked on a strategy to develop e-commerce software and the infrastructure needed to host e-commerce websites.

While getting set up to browse the Internet had been fairly simple, the work to deliver the Internet was monumentally challenging. Today, we have access to a myriad of services to assist with everything needed to create and deliver a website. In 1996, we had few such tools. We had to acquire dedicated high-speed Internet service, which by today's standards was stunningly slow and absurdly expensive (as I recall, we paid approx. $1,000/month for 256KB service). We had to buy and configure a server

computer to host a variety of services required to support Internet hosting. I had to find, acquire and configure a variety of software packages to provide these services. I had to learn the intricacies of all these services and software packages, and troubleshoot an endless series of issues, without benefit from Google et al. to answer questions, because they didn't yet exist. I also had to learn all facets of building websites and web software, which was (and still is) an esoteric combination of art and science.

Perhaps a quick lesson in Internet infrastructure would be helpful. When you want to visit a website (e.g. www.facebook.com), your browser software (e.g. Chrome) must first figure out where that website 'lives', so as to connect with it. The 'domain name system' (DNS) serves this role by providing the address of the requested web server. The web server is configured with web server software (to handle requests for various web pages) and also with programming languages and database managers to build web pages to satisfy each request. A variety of subsystems and components are needed to provide the functionality of even a simple website. The infrastructure for a website such as Facebook or Google is many orders of magnitude more complicated.

Back in 1996, I had to configure DNS software to answer the 'where is it' question for the websites we were hosting. I had to configure the web server software and an array of related components to build and deliver web pages. I had to configure an email server, to manage email for our company and for our customers, because back then the only email address someone might have was from an online service like AOL or Compuserve - not a company-specific email address (e.g. name@company.com). Almost none of the tools we use today were available in 1996, so I had to implement immature and poorly documented systems, all of which have since been superseded by more powerful solutions.

Today, you could register a domain name and create a substantial website in a single afternoon. In 1996, it took me almost two months to implement the infrastructure to deliver a simple website. By the end of January 1996, I had the Internet service installed, the web server implemented, and I had learned enough web programming to build a website for Marketplace Solutions. I don't recall what the first version of the website looked like, but by December 1996, the website looked like this:

You can still browse this version of the website (and most other websites throughout the history of the Internet) via the Wayback Machine: https://web.archive.org/. The Wayback Machine launched in May 1996 to serve as a historical archive of the Internet, and apparently it took until December 1996 for it to take a snapshot of the marketplacegroup.com website.

By today's standards, the website looks (and is) primitive. Because most computer monitors at the time were limited to 640x480 (pixel) resolution, we designed websites to fit within a 640 pixel width. By comparison, any computer or phone screen today is at least 1024 pixels wide, and we now design websites accordingly. For many years now, the primary method for formatting the layout and content of a website has been CSS (Cascading Style Sheets), but CSS wasn't supported in website browsers until late 1996, and CSS wasn't commonly used until a few years after that. Instead of CSS, we used 'tables' for layout and comparatively crude HTML attributes for styling content (e.g. fonts, colors, etc.).

I must admit that I was impressed with myself for some of the 'advanced' features I included in this website. Most 1996 websites included minimal graphics, with little visual appeal and most elements implemented via text (e.g. navigation/menus). I created a colorful/graphical look, including a spiffy 'tiled' background graphic, a variety of logos and icons, a graphical navigation menu (at the top), and even an animated GIF logo (at the bottom). However archaic this may look today, this was cutting-edge in 1996. Given that we wanted to impress potential customers with our capabilities, it made sense to demonstrate our prowess via our website.

By mid-February, I had implemented our email server, allowing our staff to use @marketplacegroup.com email addresses. Soon thereafter, we encountered the pitfalls of running your own email server - in particular, spam. As a neophyte email administrator, I had not properly considered outside threats to our server, and had not changed the default value of an essential setting ('open relay'), allowing some yahoo to send a zillion spam emails via our server. This caused major issues because many of the email addresses were invalid, causing our server to handle thousands of 'bounce' emails (e.g. 'no such user') which consumed both the CPU and disk space on our server. It took a few days to fully unwind this problem - not the last problem I encountered, but one of the worst.

To offer e-commerce capabilities, we needed to develop e-commerce software - to present products for sale and to manage the shopping/checkout process. Today, there are dozens of programming languages you can use to write website software. In 1996, early frontrunners like Java, PHP and ASP were not yet available or widely used. To that point, most major websites and most web developments were done within the Unix environment, using languages like C++ and Perl. My software development expertise was in the Windows environment using Visual Basic. As a result, I built our hosting environment on Windows servers and I wrote web software using Visual Basic. It took a bit of magic to make Visual Basic programming work on the web server, but I found a way and it worked well. Within a couple of years, Microsoft introduced Active Server Pages (ASP) which was much like Visual Basic, but integrated within the web server environment. From then through to today, I have used ASP for web development.

The first product I developed was WebOrder, a fairly simple shopping cart and checkout system. Our first customer was almost a big deal. Oakland Hills Country Club, in Bloomfield Hills, MI, was hosting the 1996 US Open golf tournament. Marketplace Solutions provided hospitality technology (point of sale equipment) for the tournament, and Oakland Hills also picked us to provide an e-commerce website to sell tournament merchandise online. After building most of the solution, we found out a couple of weeks before the tournament that the PGA had contracted with IBM to provide e-commerce services for all PGA events. As a result, we could no longer sell any PGA-branded merchandise via the Oakland Hills website. We had to remove everything except for a few Oakland Hills-specific items, for which we received only a few orders during the tournament.

Our next e-commerce website was for a very worthy but less prominent customer, my parents' Michigan products store, Michigania. We did the website at no cost, to serve as a worthy example to show prospective customers. My sister Betsy and I had a wonderful time writing the website copy in the same vein as the J. Peterman catalog. I did all the rest, once again including an attractive graphical design, another animated logo (so 1990s!) and a full-featured (for the time) shopping capability.

Prospects and Customers

As pioneers in offering e-commerce solutions, we encountered several substantial (yet unfulfilled) opportunities. For example, Craig and I flew to Kansas City, Missouri to pitch our solutions to executives at Hallmark - though apparently we didn't impress them enough to warrant further contact. We presented an extensive proposal to Olympia Entertainment (1997), to build websites for all of their venues (including Joe Louis Arena, Cobo Hall, Fox Theater, Second City, Detroit Zoo and several restaurants), with features including e-commerce, event calendars, ticketing and advertising. We lost that project because we expected to be paid for our work, but some of our competitors were apparently hungrier than we were, and they offered to do the work for little more than a feather in their cap. We presented a proposal to Michigan Farm Bureau (1998) for a massive website, to include both public and member-only content and functionality, e-commerce, advertising, and discussion forums. As I recall, we lost that project to a much larger competitor.

We also had some significant successes. In partnership with my father's former public relations company (Publicom), we built a website for the Greater Lansing Convention and Visitors Board (1996-2001), which included an extensive directory (places to go) and an event calendar (things to do). We built a substantial online store for Perani's Hockey World (1997-2001), one of the largest hockey equipment retailers in the Midwest. We built a website and online store for Big Buck Brewery (1998-2003), with its first brewery/restaurant in Gaylord, Michigan (still operating today, but under new ownership), and aggressive expansion plans (that fizzled out after building only two additional restaurants). We developed the first e-commerce systems for Purdue University (1998-2002). First for Purdue's main West Lafayette, Indiana campus, and later for their satellite Calumet, Indiana campus, we integrated payment processes into Purdue's student website to allow students to submit payment online for everything from tuition to parking tickets.

Throughout my time with Marketplace, we built websites for dozens of companies small and large (well, not that large). In 1996, only the largest companies had websites, but soon thereafter, all companies needed one. At first, many companies put up websites as little more than a statement that they were modern enough to have one. Within a few short years, having a website became essential, and having a competent website became a strategic advantage. We helped many companies to have websites earlier than their competitors and websites more advanced than their competitors.

We also built websites for many restaurants, a side-benefit of working within the auspices of Marketplace Solutions, which provided technology services to many local hospitality businesses. These included all the local Matt Prentice restaurants (Deli Unique, Morels, etc.) and the 20+ Chuck Muer restaurants scattered from Michigan down to Florida.

In 1999, we bought a portfolio of websites from a consulting firm that wanted to exit the website development business. We took on new clients, including Thermador (high-end appliances) and a dozen various Michigan-based companies. As we had no direct/personal connection with these companies, most of them drifted away over the next few years.

Also in 1999, we embarked on a major project for HealthTec.net (later renamed eClickMD) to build net.OASIS, a web-based assessment management system for Home Health Care (HHC) agencies. This project led me to develop a sophisticated web application infrastructure that I would expand and reuse throughout the rest of my career. net.OASIS allowed HHC workers to record the details of each of their patient visits so the agency could submit them to Medicare for reimbursement. Medicare's rules were extremely complicated, so net.OASIS had to enforce several layers of validation to make sure workers entered data correctly so Medicare would not reject the claims. Having such a powerful web application infrastructure proved lucrative because it allowed us to bid on some very sophisticated web application projects.

Just to be clear, when I say 'we', I basically mean 'I'. We took on a programmer from Marketplace's staff, but he could help only with basic tasks. We hired a couple of salespeople, but as with salespeople at my prior companies, it seems they could not grasp what we offered well enough to sell it. So, I did most of the selling work (with help from Craig), I wrote the proposals, I developed the software, I built the websites, I managed the website hosting infrastructure. We never got big enough to hire more staff - or just as likely, because we didn't hire more staff, we never got big enough.

Marketplace Commerce (MCommerce)

As we developed ever more e-commerce capability, and expanded our customer base, we came to believe that we were creating a valuable entity. The 'Dot-com Bubble' started in 1996, with unprecedented investments made into any company doing anything Internet related. Truly, it was a gold rush, and we worked to position ourselves to take advantage. This inspired Craig and Sam to spin off Marketplace's Internet business into a separate company - Marketplace Commerce (November 1997). I received a modest share of the company and we took on an investor who contributed funds to help us expand. But from the moment of Marketplace Commerce's creation, we were all on the lookout for a bigger fish to swallow it.

From our first entry into the e-commerce business, we sought partnerships with larger companies who might have customers needing e-commerce solutions that we could provide. We danced with Voyager.net, a Michigan-based ISP, and Ameritech, one of the regional phone companies created when AT&T was broken up (before being reabsorbed into AT&T years later). We worked with National Bank of Detroit (NBD) which had a group tasked with providing e-commerce solutions for their customers (this was the source of our project with Purdue).

We also sought entities who might either invest in Marketplace or buy it outright. We pitched the company to Rick Inatome, a well-known local investor who had built a chain of local computer stores (Inacom) and sold them for a handsome profit. He tried to pair us with another local Internet company he was coaching (1999), but there wasn't enough synergy for a match. We came close to selling MCommerce to BigNet (2000), a local Internet conglomerate that was growing explosively via acquisition, and hoping to make a fortune by going public or selling out. All the paperwork had been prepared, but at the eleventh hour, BigNet nixed the deal. Eventually, BigNet filed for bankruptcy (2003) and went out of business.

Ultimately, MCommerce must be considered a failure in that we never exploited the Internet gold rush, either by growing the company or selling it. I'll attribute this mostly to MCommerce having neither a full-time leader nor effective sales people. Craig was busy running Marketplace Solutions and gave only a fraction of his time to MCommerce. I was busy writing the software and building websites for our customers. We never found a worthy sales person. The more successful Internet startups had leaders who worked doggedly to pursue opportunities. We didn't have that.

Iacobucci & Co.

In 1998, we started a project with Millennium 321 (M321), a concept intended to profit by promoting the upcoming new millennium (2000). M321 was the brainchild of James Iacobucci and Larry Smith of Iacobucci & Co., a local marketing company. They wanted to add a slew of interactive features to the M321.com website and needed outside help to develop the needed functionality. I was skeptical that their concept would ever generate much traffic or income (and ultimately, I was right), but this project led to a much more significant relationship. In early 2001, Iacobucci & Co. bought Marketplace Commerce, so they could expand their service offerings to include website services.

MCommerce.com Domain Name

I registered my first domain name (marketplacegroup.com) on December 20, 1995. At that time, we were on the cusp of a gold rush into domain name registration. Only the largest of companies had registered domains for their companies and brands, and hordes of domain name 'squatters' started buying domain names matching company/brand names. At first, many squatters succeeded in selling domain names back to their rightful owners, but ultimately, companies used trademark law to win domain names back without paying such ransoms. Squatters also registered domains including various common words (e.g. food.com and of course sex.com) and they made small fortunes by selling domain names to startup businesses hoping to capitalize on website traffic arriving via such names (e.g. chewy.com).

After MCommerce was sold and the company name retired (2001), we no longer needed the company's domain name: mcommerce.com. It turns out that our domain name had value - because 'mcommerce' had become the nickname for 'mobile commerce', the soon-to-prosper concept of selling things via mobile phones. We received a few unsolicited offers to purchase the domain name, and within a few months, we had an offer for over $300,000! That offer didn't work out, so we kept looking for a buyer, even employing my old pal Tim Taschler to lead the search. Then, the 'Dot-com Bubble' burst, and Internet company valuations plummeted, leading to much lower demand for domain names.

It took over a year, but we finally found a buyer, who leased the domain name with an option to buy it. A few months later, they stopped making payments, so we ended the agreement and renewed our efforts to sell the domain name. Over the next few years, we received several offers, most of them laughably low. Finally, in 2008, we received a serious offer, and we sold the domain name for $9500 - not the windfall we had hoped, but a nice bonus nonetheless. Now in 2023, I see that mcommerce.com is owned by Shopify.com, one of the leading e-commerce companies, what MCommerce aspired to become.

Technology

My employment with Marketplace Solutions lasted nearly two years and with Marketplace Commerce just over three years. Throughout that time, I worked within Marketplace Solutions offices, and due to my expertise, I served as tech support for the entire office. I arranged our first dedicated Internet connection (256KB) and later upgraded it to a 'full T1' (1.5MB). I helped acquire and support their voicemail system, which back then was a computer stuffed with telecom boards and sophisticated phone/voicemail software. I acquired powerful servers for website hosting and did everything needed to keep everything running 24/7/365. This included carrying a pager, both to handle calls from staff and also from a service that monitored our servers.

Today, hosting lives in the 'cloud' - that is, racks and stacks of thousands of computers managed within colossal data centers. Cloud hosting is convenient in that you don't have to manage the servers and connectivity yourself - the hosting company does it for you. From the start of Marketplace's Internet business, we had to provide our own hosting solution, which required fast Internet service, hefty servers, and a variety of software to make everything work. Our customers expected their websites to be accessible nearly 100% of the time (and rightly so), so we needed to keep our servers running and online nearly 100% of the time. We used a monitoring service to track whether our servers were accessible. If not, the service sent a message to my pager, which for a couple of years was a constant presence on my belt. Soon thereafter (1998), I carried a cellphone, so I added my pager to the museum of the obsolete (I got rid of it).

If the Internet went out (a regular occurrence back then), I would maintain a vigil with our Internet provider until it was fixed. If the power went out, whether day or night or weekend, I had to get to the office and start up a generator to power our servers and Internet equipment - definitely not my favorite part of the job. I remember once that the lights in the office dimmed, and we went outside to watch a high-voltage line sag slowly lower until it touched the local electric line and zapped all the power in the

area for more than a day. Fortunately, Marketplace staff often helped me with generator startup and refueling.

Career - 6. Iacobucci & Co.

James Iacobucci and Larry Smith were partners in Iacobucci & Co., a marketing company in Birmingham, Michigan. In early 1999, they hired Marketplace Commerce to add interactive functionality to their Millenium 321 (M321) project, a concept intended to profit from the upcoming new millennium (2000). James and Larry had visions of great fortunes to be gained by taking M321 public, but that would never happen - in fact, just like the overblown Y2K apocalypse, M321 never gained much audience and disappeared soon after the start of the new millennium.

Within a couple of months of working together, James and Larry began seeking a closer relationship between Iacobucci & Co. and Marketplace Commerce, first by offering Marketplace a share of the M321 company in return for website development services. We wisely declined that offer and instead charged them hourly fees for our work on the project. However, James and Larry and I forged a relationship, and they started to include me in their project pitches, as the 'web department' for Iacobucci & Co. One of our first joint projects was Walsh College, where we collaborated on a website redesign - they did the design, and I implemented it on the Walsh website.

By late 2000, Marketplace Commerce had exhausted its attempts to gain a windfall by selling out to a bigger company. Craig and Sam's main hospitality technology business was booming, and they increasingly focused their attention there instead of on the Internet business. James and Larry wanted to expand Iacobucci & Co.'s services to include website design, development and marketing. Eventually, we all came to an agreement to sell Marketplace Commerce to Iacobucci & Co. (January 2001). I was the only Marketplace Commerce employee to make the move - into Iacobucci & Co.'s office suite above the 220 Merrill restaurant in downtown Birmingham, Michigan. I experienced a lovely culture shock, moving from the bland, fluorescent-lit, cubicle world of Marketplace Solutions into the artistic Iacobucci office, with dub/chill music wafting on restaurant-scented air blowing through windows open to the hip Birmingham walkways below.

The honeymoon was short-lived, as less than two months later, Larry left the company (due to transgressions I'll omit). James was a gifted and artistic marketer, having left a high-end agency to start his own marketing company. Unfortunately, he focused far more on his art than on running the company, and Larry's departure put the company in a difficult financial situation. Although we stayed afloat, we seemed to always be on the edge of disaster.

Key Customers

James and I worked well together, and he pushed me to better understand marketing and the visual arts. We produced some beautiful websites, including one designed entirely in Flash, a now-obsolete graphics/animation environment used to create more visually interactive websites. Learning to design and code in Flash was one of the most difficult challenges I have taken on, in part due to the heavily graphical nature of the work, but also because Flash includes a timeline along which elements evolve, much more like making a movie than a website. Fortunately for me, there were many problems with

Flash-based websites, including major security flaws, so Flash websites quickly disappeared in favor of more traditional coding methods.

Marketplace Solutions and Iacobucci both had strong connections in the world of golf and country clubs. Craig and Sam had both joined Oakland Hills Country Club (OHCC), the poshest club in the area, where they schmoozed with the local business elite. Iacobucci shared office space and business relationships with a couple of avid golfers who were also OHCC members. These relationships generated several opportunities for us, including a project to build a website for The Wyndgate, one of the grandest new golf clubs in the area. The Wyndgate was built by Larry Winget, owner of an automotive supply company, with little regard for cost, resulting in fabulously designed buildings and golf course. It may have been built with ill-gotten money (Winget was sued for siphoning millions from his own company before it declared bankruptcy), but it was exceptional. Fortunately for us, they also had the money and design-sense to produce a fabulous website, which I used as a reference site for years.

We did another project for Kinsale, a golf club to be built in Naples, Florida, funded by Tony Soave, a Detroit-area entrepreneur (with questionable relationships including flawed Detroit mayor Kwame Kilpatrick). The golf course was to be designed by former PGA professional Tom Weiskopf (to whom I loaned a jacket for a photo shoot at OHCC). As this club was to be as extravagantly built as The Wyndgate, Kinsale also splurged on Iacobucci's design services. However, the project collapsed before they built anything other than a website and some promotional materials, and the concept disappeared.

Our work with The Wyndgate led to another huge project, a web application to help companies manage the legalities and paperwork required to bring temporary workers to the US from Mexico via the H2B visa program. The Wydngate brought in dozens of seasonal workers to build and maintain the golf course, which included building miles of brick paver cart paths in addition to typical golf course construction. They became skilled in navigating the many issues encountered via the H2B program, both with getting the best workers to return each year and getting the workers successfully into the US to work. They spun off a separate business (LLS) to provide these services to other companies needing seasonal workers. I built a sophisticated web application for LLS to allow them to register thousands of workers in Mexico and to manage requests of their US customers to get H2B visas for the workers they needed.

A Sad Ending

Larry had led Iacobucci's sales efforts, finding and following leads, while James focused more on marketing strategy and design for clients. After Larry left, James struggled to generate the quality leads that he and Larry had generated together. By early 2003, despite my website clients supplying substantial revenue, the company was unable to pay my full salary.

I was forced to consider leaving Iacobucci to find a better-paying job, but I also considered an alternative: starting my own company. If I left Iacobucci, he would have been unable to complete our ongoing website projects (he did design, not programming). Nor would he have been able to do the needed maintenance work for the numerous websites we managed. While Iacobucci's total revenues could not support multiple salaries plus fairly hefty expenses (like Birmingham office rent), my website

work was generating enough revenue to pay me a decent salary plus modest expenses. It took a bit of arm twisting, but I convinced James to let me take the website business off his hands in exchange for me forgiving much of the money he owed me, plus a small share of my future revenues.

In April 2003, I created Direct Knowledge, and I spent the remainder of my career working as a sole practitioner.

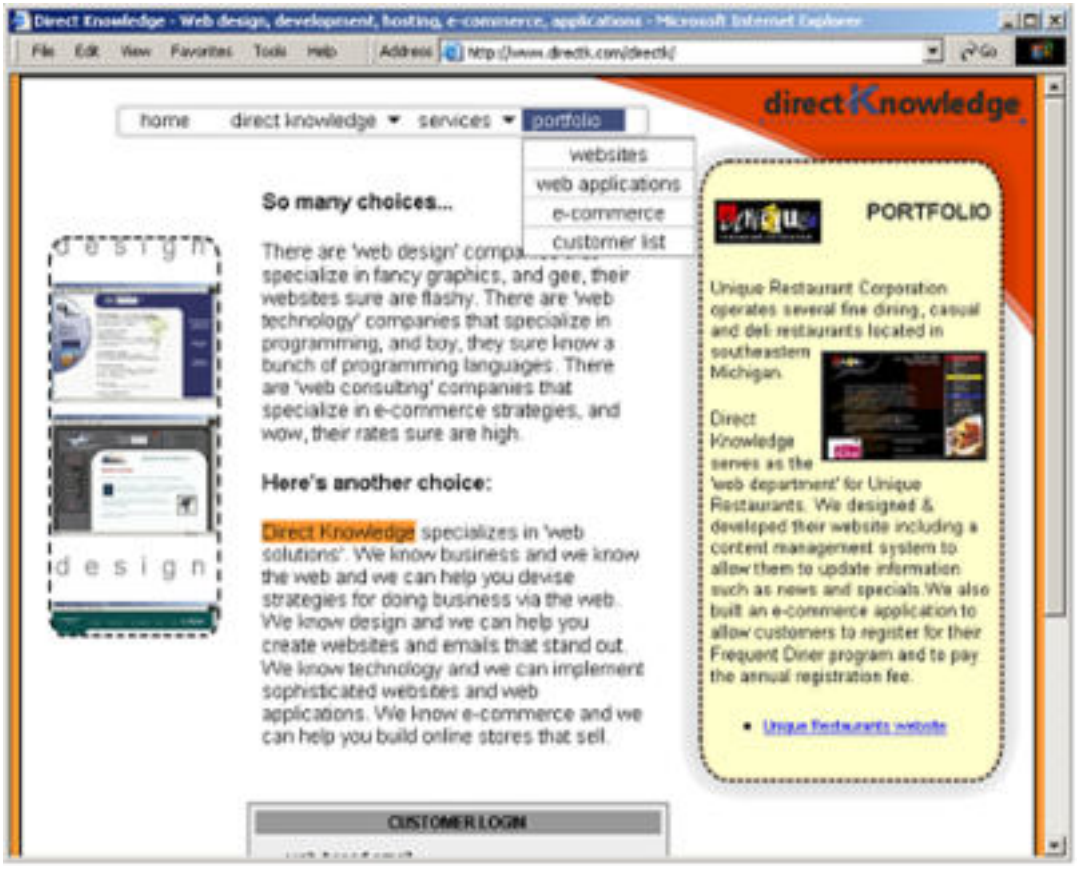
Career - 7. Direct Knowledge

As my situation with Iacobucci & Co. deteriorated, I worked on plans for a new company that would take over Iacobucci's website business - me taking over the business that I had brought to Iacobucci via its acquisition of MCommerce. This was a complicated situation because Iacobucci still owed MCommerce more than 3/4 of the purchase price, and because I was a part owner of both MCommerce and Iacobucci & Co. My MCommerce partners wanted their money, but Iacobucci had little ability to pay it, and if I took over the website clients, Iacobucci would lose most of the assets for which it was paying.

I consulted my favorite attorney (thank you, Joe Hemming!), and we all negotiated for a few weeks. In the end, I could not take with me any of the assets that Iacobucci had bought from MCommerce. That included the e-commerce software I had written (WebCat, WebOrder, etc.), and also the servers we used to host websites and email for our clients. What I could acquire was a couple of computers, the software I had developed while I worked for Iacobucci, and the client relationships Iacobucci had with a couple dozen website services clients. While it may sound like a travesty that I had to leave the e-commerce software behind, the only real travesty was that Iacobucci had purchased e-commerce software with quickly dwindling value. The web applications marketplace was exploding with new technologies and software, and Iacobucci's focus on website services rather than website software resulted in our software falling behind.

My short-term plan for the company was to continue the website design, development and hosting work I had been doing with Iacobucci. Longer-term, I planned to develop an organizational knowledge management system, which would marry my website expertise with my prior knowledge management expertise. I sought a company name based on the keyword 'knowledge', and for which I could find a good/short domain name. Through a lengthy search, I found three good domain names: activek.com, directk.com and onlinek.com, each of which might work well for the software and services I planned to offer, and from these, I decided that Direct Knowledge (DirectK) was the best company name.

I started DirectK on a shoestring. Iacobucci had not paid the bulk of my salary and expenses for a few months, and I took on sizable financial responsibilities via purchasing the business from Iacobucci. Having little money to invest in the business, I did everything I could on my own, and found inexpensive options for everything else. I designed a logo, letterhead and business cards. Graphic design is not my expertise, but rather than pay a graphic designer, I did it myself (and I'm still happy with the results). Of course, I built a website for the business, though I found it challenging to quickly create a design and content worthy of the story I wanted to tell to prospective customers. I set up QuickBooks so I could send invoices and get paid. As my only employee, I answered the phone, made the sales calls, wrote the proposals, did the client work, billed the clients and, by necessity, did everything else that needed to be done.



At first, I thought I could work from home, but within a day or two I realized I could not share space with my unruly sons (aged 15, 13 and 11). Our house did not include a separate office, and our household chaos disrupted my work in the dining room. So, I searched for inexpensive office space, finding a few small offices in very bland office buildings. Most of these would have been fine, but I had been happy with the cool vibe of my office at Iacobucci, and I hoped to find something equally satisfying. I found what I wanted in a decrepit building in downtown Farmington, Michigan, a sizable 2nd floor office, with floor-to-ceiling windows overlooking woods, a river and a park, and with the bonus that it cost less than any of the others. The landlord replaced the ancient carpet, and Lisa and I painted the walls, quickly transforming a neglected space into something quite nice. I bought a desk and acquired some tables and guest chairs from a storage unit holding remnants from my father's former Publicom business. I warned all of my visitors to ignore the shabby entry into the building and up the stairs - my office made the trek worthwhile.

While we were preparing the office, I left most of my equipment and files in my Iacobucci office. One day I got an email from James requesting that I pay rent and expenses totaling over $600/month. Given the financial hell he had put me through, I did not take this well. That evening, I took my boys to Iacobucci's office, and we cleaned my office out. I think it shocked them to be involved in such an abrupt and clandestine mission. When James came in the next morning, my office was empty, and a curt email was in his inbox, letting him know I was paying rent and expenses elsewhere and would certainly not be paying any rent to him. He seemed humbled by my response, and we maintained a civilized relationship thereafter.

When starting a new business, you encounter many needs that you wouldn't even consider at an established business. I ordered new phone and Internet service and I bought an inexpensive phone and voicemail system. I bought a multifunction printer with a scanner (to digitize client marketing

materials and photos for use in websites) and fax functionality (it would be several years before I could finally abandon the fax machine and its separate phone number). I set up a new bank account and ordered checks so I could pay my bills. I designed and installed signage for the new office to guide visitors to my door. The list goes on. Within a few months, I had everything I needed, and I settled into the rhythm of running my own website business.

A Big (Lost) Deal

The most significant driver of my decision to start the business was my largest client, LLS. For many months, I had been doing substantial work for LLS and I had ongoing projects that would last several months longer. During my negotiations with Iacobucci and MCommerce, I also negotiated a deal with LLS that would assure me of continued billings in exchange for a lower hourly rate for LLS. The other clients I would take on could not supply sufficient billings to sustain the company, so it was imperative that I retain the LLS billings. Yet less than one month later, LLS informed me they would not sign the negotiated deal and that they would transition to another web developer. It was a crushing blow. Had I known this was coming, I would never have started my company.

Despite months of angst that followed, losing LLS was perhaps the best thing that could have happened for me. With LLS on board, I would have focused almost exclusively on their projects, and would have done little work to find additional clients. Having lost LLS, I became extremely motivated to find new work, and for many years thereafter, I continued to devote much of my time to marketing. Fortunately, my work for LLS continued for a few months, and while my early billings were meager, I survived until I found more work.

I started networking in every direction I could. I sought work via my former partners (despite the failings of our business together). I prodded my customers to see if they might need additional work. I investigated the local chamber of commerce and several similar marketing/networking opportunities. I found Business Networking International (BNI), with thousands of chapters worldwide, and each chapter composed of various professions, all working to generate business for each other.

A BNI chapter can have only one company for a given profession, so I had to find a chapter that did not already have a website services company, which turned out to be much easier for me than for professions like realtor or insurance agent. Chapters vary in the number and quality of the businesses represented. I visited one chapter, which met in a noisy cafeteria, and they seemed casual in their approach to networking. I was unimpressed. However, I was also needy for business opportunities, so I tried another chapter. They met in a nice conference room and seemed very serious about generating business. I joined their chapter and spent nearly 15 years participating in a series of BNI chapters. Via BNI, and some of the exceptional people I met there, I generated most of my biggest client projects at DirectK.

Moving to the Cloud

From my start in the Internet business, I built and managed my own website hosting infrastructure (Internet service, servers, software, etc.), because there were few alternatives. Within a few years,

many companies offered 'managed website hosting' services, whereby they provided the needed infrastructure, which you could manage remotely. As I have described, keeping all that infrastructure fully operational and accessible was challenging, and to be honest, annoying (things seem to break down at the most inopportune times).

Starting in late 2001, I hosted new clients via a managed hosting service (MaximumASP), and over the next couple of years, I migrated all of my existing customers there. In a sense, this meant I was moving into the 'cloud'. This did not end the problems I encountered in keeping websites up and running. Even in the cloud, computer hardware and software still fail. But in the cloud, it became someone else's job to keep the computers and the Internet running, which was a welcome relief to me. By 2004, I had moved everything off the original MCommerce servers (still housed at Iacobucci) and they became obsolete and unused.

When I started writing website software, I used Visual Basic and a bit of wizardry to get the software to run on the web server. Again, I chose this path because there were few alternatives. As web servers and web software quickly evolved, this technique became ever more awkward, suffering issues of slow performance, poor support, etc. Software providers soon offered web-oriented programming environments which solved most of these issues. For me, this meant converting my software to Microsoft's Active Server Pages (ASP) programming environment, still based on the Visual Basic language, but fully integrated within the web server environment. Because ASP was supported within any Microsoft-based web server environment, this meant I could easily move my software from my own custom-built infrastructure to any of the managed hosting providers.

Website Software

ASP also opened up a wealth of web software that I could deploy for my clients. Rather than having to develop all the software myself, I could acquire full-featured software from other software providers and then customize it to address my clients' specific needs. For example, I used an e-commerce package (CandyPress) as the foundation for dozens of online stores. CandyPress offered features far beyond what I had developed in my WebCat and WebOrder software, so it served as a capable starting point for my projects. Because I had full access to the source code for the software (and rights to tinker with it), I could customize the software for my clients, and I could also develop custom extensions of the software. I could rely on the software developer to fix bugs, add new features, and modify the software to keep up with evolving web standards. I then had only to update my own extensions to the software.

The early 2000s saw a shift in how website content was managed. In my first few years creating websites, I built ‘static’ websites - with each web page coded in HTML (the coding language used by a web browser to present a web page). My clients sent me requests to update the verbiage and/or graphics on a page, and I made the updates for them. As websites became more integral to business operations, and as people became more website savvy, the industry shifted to Content Management Systems (CMS) via which non-technical people could update website content using an online system. At first, I developed simple systems to allow my clients to update verbiage on their website pages. Later on, I implemented systems to give them almost complete control of their website content.

In 2004, I built an extensive 'portal' website for Business Advisor Online (BAO), a startup company planning to offer information and advice to small businesses. The BAO website required paid membership for access to curated collections of articles and links, categorized by all the facets of running a business (e.g. finance, technology, HR, etc.). BAO also offered surveys/polls, discussion forums, bulk emails, online advertising, and more. But the key to the entire project was that BAO would manage the content, not me. I found a 'portal' application (AspApp) that offered much of the needed functionality, and I customized and extended the functionality to address BAO's requirements. While AspApp was a capable system, it had many limitations and a few flaws, and this led me to consider developing my own portal software.

In 2005, I rebuilt the website for Meadowbrook Country Club (MCC) in Northville, Michigan. MCC's existing website allowed them to make some updates to the website, but they wanted more functionality and more control. MCC wanted to offer its members a members-only website, via which MCC could present articles, photos, calendars, and rosters to MCC members. Rather than basing this website on a package like AspApp, I used this project as inspiration and impetus to develop my own CMS.

Active Knowledge (ActiveK)

I had been building Membership Management Systems (MMS) since the mid 1980s, so I understood the requirements for managing member information, membership registration and renewal, members-only content, calendars, rosters, bulk email, etc. I had been building and working with CMS software for several years. I took everything I had learned about CMS/MMS software, and tackled the requirements of the MCC project, to build version 1 of ActiveK (a.k.a. Active Knowledge), a CMS/MMS software package that I would implement for dozens of projects over the next 15 years.

At first, ActiveK was a useful but very basic CMS. It satisfied MCC's requirements (they used ActiveK for nearly 10 years), but it had substantial limitations. However, it gave me a leg up on each of my subsequent project proposals. The functionality already included in ActiveK allowed me to tackle ever more sophisticated website projects, at prices below what my competitors charged. Each project then demanded that I develop more functionality, which I built generically into the software, allowing me to tackle yet more sophisticated projects. Eventually, ActiveK became a full-featured CMS, with a built-in MMS, and even online store functionality.

To build ActiveK, I borrowed concepts from AspApp and also from my prior works, going all the way back to FM80 and Wincite. I started with the web application infrastructure I had been advancing since my earliest web projects. From AspApp, I built a page layout (templating) function to allow users to create elaborate page layouts, pulling articles, calendar entries and online store items into various sections on a page. From FM80 and Wincite, I built a category management system to allow flexible organization of categories and articles for even the largest of websites. From CandyPress, I borrowed techniques for credit card processing and integration with shippers (e.g. UPS, USPS, Fedex). For each customer, I took their unique requirements and extended the ActiveK software to address them. I even borrowed from Google - after I learned the basic methods used to build a search engine, I built my own (simple) search engine within ActiveK.

Through the years, I implemented dozens of websites via ActiveK, and I used ActiveK as the framework within which to build a variety of sophisticated web applications. And, I continued to improve the ActiveK software both to add functionality and to address evolving technology and security trends.

Website Marketing

Google started up in 1998, and it proved superior to several existing search engines at generating useful results. Within a few years, Search Engine Optimization (SEO) became an essential tool to help websites appear near the top of search results. A battle ensued, pitting marketers trying to game the system to gain high search rankings, versus Google trying to deliver useful search results. SEO hotshots would find new techniques to get high rankings, with tricks like stuffing keywords into pages as invisible text or including competitors' company and brand names as keywords. Google (and the other search engines) discovered these tricks and modified their software to thwart them.

As a website developer, it became essential for me to learn SEO and include SEO techniques in my customers' websites. While I never became an SEO guru, I generated first-page search rankings for most of my customers' primary keywords. I also built SEO capabilities into ActiveK, so that my customers could produce search-optimized pages via ActiveK.

In the early 2000s, Google introduced their AdWords system, via which Google displayed Pay-Per-Click (PPC) ads along with search results. Another battle ensued, as companies bid ever higher amounts (per click) to ensure their ads were prominently displayed. As with SEO, marketers learned techniques to get the biggest bang for their buck. SEO and SEM (Search Engine Marketing) became its own industry, with a slew of companies specializing in all facets of website marketing. I dabbled with AdWords for a few customers, but the work is mostly outside the context of website development, so I left SEM work to the experts.

I Made a Few Mistakes

I first built a website for Reynolds Water Conditioning (RWC) in 2008, using a combination of my early CMS software (pre-ActiveK) and CandyPress for an online store to sell water filtration equipment and supplies. In 2009, I replaced the CMS software with ActiveK, keeping CandyPress for the online store. In 2012, I redesigned the RWC website, and I replaced CandyPress with the new online store functionality that I was adding to ActiveK. However, the ActiveK store functionality was not complete, nor thoroughly tested. For several months after the new site went 'live', we encountered a variety of bugs, including some that prevented RWC customers from placing orders. RWC was justifiably annoyed. These issues were compounded by issues I was having with my hosting provider Alentus (see below). RWC stuck with me for over a year, but eventually, they tired of the continual issues and they had another company build them a new website. Because DirectK was such a tiny software development company (just me), I had to push the envelope by developing software quickly and with only cursory testing. Most of the time, I got away with it. For RWC, I failed.

In 2016, I started a website project for Michigan Photographic Historical Society (MIPHS), a small volunteer-driven organization devoted to the history of photography and photo equipment. As I did for

most of my customers, I took control of their domain name (miphs.org) so that I could manage domain name renewal and control domain name usage. Due to a cascade of errors, I neglected to renew the domain name. I had thought it was set to renew automatically (as I configure all the domain names I manage), but when they transferred the domain name to me, auto renewal was not enabled. I then failed to notice warnings that the domain name would expire and after its renewal date, it expired. Unfortunately, this happened before my domain name registrar (GoDaddy) implemented procedures to prevent domain name loss due to expiration. Instead, GoDaddy offered a system via which you could 'backorder' a domain name, such that if the domain name became available, the backorder process would automatically purchase it on your behalf. More unfortunately, the buyer who backordered the domain name was a company that bought thousands of expired domain names with the purpose of monetizing any traffic that came to those websites. Some such companies present advertising on the websites and profit from visitors clicking on the ads. Extra more unfortunately, the company that bought the miphs.org domain name was a Russian pornography site hoping to profit by selling porn. I can partially attribute my lack of diligence to my inner turmoil after my mother's death a month earlier. But that's an insufficient excuse. I messed up. It was horrific for MIPHS to deal with their members after encountering the porn site.

I immediately purchased another domain name for their use (miphs.net) and I hosted their website at no charge for several months. MIPHS threatened legal action, but they only sought to get back the deposit they had paid for the project. Obviously, I gave them what they asked, relieved that they hadn't asked for more. Eventually, they took over the new domain name and the website. Six months later, I discovered that miphs.org had once again become available (apparently, it didn't generate much traffic for the porn site). I let MIPHS know, and they repurchased the domain name. It was an ugly chapter in my career, but I was glad that it worked out OK.

I started hosting websites in the cloud with MaximumASP in 2001. At first, I was happy with their service, but the way it was configured caused issues for me in hosting multiple client websites on their servers. By 2006, I needed a better solution, so I moved website hosting to Alentus, a similar hosting provider, but with a better solution for hosting multiple client websites. While I encountered many issues with Alentus, mostly about downtime, whereby our websites to be down for a few minutes or even a few hours, Alentus worked out fairly well for me. In 2014, one of the Alentus servers experienced a series of errors that resulted in many of my clients' websites being inaccessible - for three days. The server equipment and software had become outdated, and I should have moved the websites elsewhere before such a calamity could have occurred. As a result, I lost a couple of clients (including RWC), and I came close to losing a few others. I quickly moved all my clients' websites to a new hosting company (LiquidWeb, in Lansing, Michigan) and I have had no such problems since.

But I Avoided Most of Them

I mean no offense to my customers, but they don't know what they want. They understand the business problems they hope to solve with software, but they (mostly) don't know what software they need to solve the business problems. Unfortunately, they think they know what they need, and many software developers will build it for them. All too often, they build the wrong thing or they build it the wrong way, resulting in software that doesn't actually solve the business problems. Such projects are often

expensive, because the software is often more complicated than needed, because scope-creep causes a cascade of expensive add-on work, and because sometimes the project is so off-target that it must be scrapped and restarted.

I avoided most such issues by conducting a thorough requirements definition before starting each project (a skill I first learned at Price Waterhouse). For more elaborate projects, I convinced customers to pay for a separate requirements definition project, resulting in detailed specifications, plus my proposal to address them. I worked with customers to understand the business problems they needed to solve (a skill practiced throughout my entire career), and we worked together to devise good solutions. Getting the requirements right means that I almost always build the right solution - software that solves the business problems, and that is intuitive to learn and use.

Having solid requirements also allows me to quote projects accurately. Many of my competitors do not get proper requirements and therefore quote projects based on flawed or incomplete project scope. Customers become annoyed when midway through a project, a developer tells them that the project quote did not address needed features and it will cost extra to address them. I rarely encountered that problem.

All software has bugs. You can search for bugs and fix the bugs, and yet still your software will contain bugs - lots of them. It may take a long time for the right situation to arise to reveal a given bug, and when it does, it may take a long time to track down the bug because 1) bugs can be very good at hiding and 2) over time, you lose the intense familiarity you had with the code when you first wrote it. To minimize bugs and their impact, my approach is to fix it, fix it and then fix it. First, you need to find the bug and fix it, a process that is often challenging, both the finding and the fixing. Second, search your entire code base to find any similar bugs and fix them too (often a buggy technique is repeated elsewhere in the code). Third, fix any other stupidities you find along the way, because while you track down the bug, you regain familiarity with the code, and you often see stupidities that would not otherwise stand out - so fix them too. This approach leads to much better code and fewer bugs, and this leads to happier customers and fewer late night debugging sessions.

Perhaps the easiest way to avoid problems, or rather to keep small problems from becoming large problems, is to simply own them and resolve them. If it's broken, I fix it. If a feature should have been in the software, I add it. I make it my problem, not the customer's. It may cost me time to fix things, but it keeps my customers happy and gives them confidence in me - which often leads to add-on projects.

So What If I'm a Luddite?

Throughout my career, I have been reluctant to move to new technologies. At Quest, I resisted our move into Windows-based software, because we had so much invested in building and supporting our existing software. I was wrong, because Windows was the future, and while our move to Windows was challenging, it was absolutely worthwhile.

For many years now, I have resisted moving from Active Server Pages (ASP) programming to any of several more modern programming environments. My rationale is mostly based on inertia - I have

developed a wealth of software tools in ASP that I would need to recreate in any other language I might choose to learn. While ASP has shortcomings (including becoming obsolete), it continues to serve me and my customers well. The software industry has produced a never-ending supply of new and better programming languages. While knowing more modern languages would help me get a programming job elsewhere, it wouldn't necessarily help me produce better software for my customers, so I stick with ASP.

As with many of the technologies I have worked with, I started work on them before they had become mainstream, before quality solutions were readily available. This was also true with CMS software. As my customers demanded CMS solutions (early 2000s), there were few alternatives available to them. WordPress (and several of its brethren) was in its infancy. So I built my own CMS. Over the ensuing years, WordPress (and its brethren) became increasingly capable and popular. Had I started in website development in 2008, I would likely have used WordPress to build websites, and I would have developed custom applications via WordPress extensions and widgets. Instead, because of my know-how and the capability I built into ActiveK, I continued to use ActiveK instead of WordPress.

While we're discussing WordPress, I'll let you know my opinion of such CMS software (my rant begins here). WordPress has become an increasingly capable platform, and because of its popularity, it has accrued a massive library of design themes and programming widgets and a vast population of WordPress consultants, which all together can help you build a stunning array of websites and web applications. However, as a website developer who has had to take on and support a few WordPress websites, I despise WordPress. The good news is that you can shop a vast marketplace of design themes and widgets to apply to your WordPress website. The bad news is that these themes and widgets were developed by different developers, using different programming techniques and offering different user interfaces. A given WordPress website might include one or more design themes and a dozen or more widgets (some of which may be disabled, for reasons unknown). It can be a nightmare to figure out how to make updates within such a cobbled-together website. It can be a nightmare to keep all these components up to date, and to avoid bugs and security issues. I am not a fan.

At this point in my career, the website development world has left me far behind. My choices have led me to obsolescence. I don't know the programming languages used by modern developers, and I'm no longer interested in learning them. I don't know the intricacies of modern (cloud) hosting platforms like Amazon Web Services and Microsoft Azure, and I see little value in exploring them. I don't know modern CMS platforms like Wix and Weebly and Squarespace, and while I may tinker with them some day, I have no plans to become an expert. I am fortunate that Lisa and I have sufficient savings to allow me to retire, because it has become time for me to leave the website business (which I am doing, as one by one, my customers leave me for a more modern solution).

Partnerships

I sought customers via my website, the yellow pages, and business directories (e.g. Chamber of Commerce), and a few of my biggest customers found me this way (e.g. Fourgreen and Edward Rose & Sons). But most of my customers came to me via networking.

Through my participation in BNI, I shared leads with many great business people. I first met Mike Morris of The Providers (specializing in computer hardware, software, and networking) soon after I joined BNI in 2003. Over the years, he and I shared many leads and worked together on many projects. He brought me a few of my biggest customers (StraightLine, Rex Materials Group, Weaver Media) and I count him as a friend.

I met Emil Izrailov (mortgage loan officer) early in my work with BNI, and he brought me my largest customer (OfferSubmission). Emil is a great guy and I have a story to demonstrate. One morning, we left a BNI meeting, and I found my car had a flat tire. I started digging in the back for a jack and the spare, but Emil came up and said, “let me do it.” I said that I could handle it, but Emil insisted. While quickly changing the tire, he told me his parents had emigrated from Russia to the US, and his large extended family then chain-migrated to the US to join them. As each family arrived, with little more than their luggage, the US-based family would house them and help them start their new life in the US. This included helping them with transportation, which usually meant crappy used cars. Emil and his brothers became the road crew, fixing the cars and changing flat tires. He told me he had changed over 100 flat tires, so he could change mine much faster and easier than I could. Thanks Emil, for all you do!

Customers and partnerships can come from anywhere - that is the nature of business networking. I got leads via people in my neighborhood, people in my church, even via my children's friends. One of my biggest projects (Florists) came via Mike Etienne, a fellow swimmer on my masters swim team. I helped Mike with several projects, including helping integrate his florist software with florist websites, and even (briefly) serving as his backup when he went on vacation.

I also value my partnership with two graphic designers, Chris Van Wyck and Becky Faber. I care about graphic design and I can envision what my customers' websites should look like. But I cannot begin to produce the brilliance I might envision and I never gained sufficient expertise with graphic design tools (e.g. Adobe Illustrator) to create anything truly attractive. For many websites that I built, the website didn't need exceptional design, and the project fees wouldn't pay for graphic design work, so I did the basic design work myself, sometimes using a pre-built design template. But for more substantial websites, and particularly for those needing visual appeal (e.g. country clubs and restaurants), I subcontracted the graphic design work to Chris or Becky. On so many occasions, they made me look good, and I thank them for their creativity.

My Offices

I loved my Farmington office with its tranquil, verdant views and sounds of birds and burbling water from the river passing below. A small birdhouse hung just outside my window, and I watched generations of sparrows live there. Each spring, I saw the parents toil to build a nest inside, and then some days later, I heard the cheeps of baby sparrows, as the parents ceaselessly brought food to the babies and removed waste from the nest. I only witnessed 'fly away day' a couple of times, and it was exceptional theater. It took most of the day, as each baby bird (now near full size) stepped onto the perch of the birdhouse, and like a young child on a diving board, stood there finding the courage to take the leap. The parents sat nearby, chirping encouragement, sometimes even swooping at the birdhouse to force a reluctant child to take that first step. Some of the young birds took right off and delighted in flitting

amongst the branches. Others were more cautious. But by the end of the day, all of them had flown away.

Also entertaining was f***ing day. My office windows had a ledge running the length of the windows, about two feet up from the floor, just below the birdhouse. Before there were baby sparrows, there was f***ing. Lots of f***ing. All day long, sometimes for a couple of days, the male (or possibly males) would hop onto the female, and give her some loving. Again and again, all day long. It was mildly entertaining, but distracting. One day, I had customers visit my office on what happened to be f***ing day. It was disturbing (and hilarious) to hold a meeting while birds went at it just outside the window.

The office backed up to a tree-lined river with a park beyond the river. I couldn't see much of the park, but I could often hear the sounds of people playing and picnicking. Along the river, on both the office side and the park side, I found large swaths of black raspberry bushes, one of my favorite delicacies. For a few weeks in late June and early July, I picked berries almost every day, some to snack on right then and there, and some to take home for a higher purpose like cobbler or ice cream. One plant sat on the crest of the river bank, with runners extending 10+ feet down to the edge of the river below, and I could nearly fill a container with luscious berries from that plant alone.

In 2011, after our boys graduated high school, Lisa and I moved to a new house in Milford, Michigan. While our Farmington Hills house had no separate office, and had been filled with unruly boys, the Milford house was larger, and it had a separate office, and we no longer shared space with unruly boys. A part of the financial rationale for buying the Milford house was that I would no longer need to pay for a separate office, so I sadly left the Farmington office behind, and I moved everything to Milford. Soon after moving to Milford, I stopped using the office for everyday work, instead opting for the sofa in the living room (where I still do most of my work today). Working from home has been beneficial for me, because I used to sit in an office working at my computer for 10 hours a day, but in Milford I could take regular breaks to work in my gardens or wander the woods, a much healthier work environment.

Career - Appendix A - A Lifetime of Technology

As I write my career memoir, I find myself explaining an array of technologies that, during my relatively short lifetime, have become so obsolete as to seem improbable and foreign.

For example, I might describe how at one time, we relied on 'fax machines':

"A fax machine uses an optical scanner to create digital images of printed pages and then sends the images via telephone line to another fax machine which receives and prints out the images. We used fax machines to send documents and forms instantly, a great improvement over sending such documents by mail or Fedex."

It occurs to me that mere decades from now, even these descriptions will prove insufficient. My description refers to 'telephone line'. Soon enough, teenagers may ask, "what's a telephone?", unable to recognize the relationship of a 'telephone' to whatever computer/communication appendage they may use. Already the concept of 'telephone line' has become archaic as homes no longer have 'land lines', and the only wires coming into a house are electric and Internet (and soon enough, even those wires may disappear). I can't even explain my explanation without referencing obsolete technology (e.g. 'land line').

Please understand that I was 14 years old before my TV offered more than 3 channels. I was 22 the first time I used a personal computer, 35 the first time I browsed the Internet, and 49 when I acquired my first smartphone. Today, most three-year-olds can easily navigate a smart device to watch videos streamed via Internet. What has been new for me throughout my life is commonplace for toddlers today.

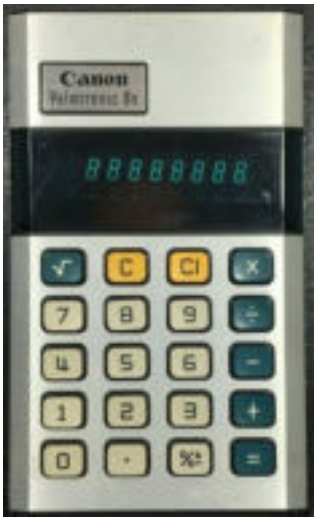
I have seen videos of teenagers struggling to understand the concept of a 'pay phone' or how to use a 'dial phone'. Vestigial concepts like 'dial a number' and 'hang up the phone' litter our language. Surely, over time, most of these concepts will disappear. Soon enough, most of the technologies we use today will suffer a similar fate.

So, I feel obligated to go full-on 'old man', and describe the great technological progress I have seen in my lifetime. Please forgive my inevitable use of obsolete references to describe other obsolete references. I'm doing my best here.

Computers

Much as kids born in the 1990s have never known a world without Internet, or kids born in the 2000s have never known a world without smartphones, I was born into the age of computers. My elders used the slide rule to make calculations for algebra and calculus classes. In the early 1970s, mine was the first generation of students to use handheld electronic calculators instead of the slide rule - in fact, I never used a slide rule, something inconceivable to students only a few years older than me. Instead, I

used a calculator, much like the basic calculator app on a smartphone, but as a separate device a bit larger than a smartphone.



I first encountered general-purpose computers in 1978 when I took a programming class at the University of Michigan and I first encountered personal computers in 1982 at Central Michigan University (you can read about my early experiences with computers in chapter 1). I started carrying a portable (aka 'luggable') computer in 1985 - a 30+ pound beast that I carried by its handle as its shape and weight defied carrying by shoulder strap. Over the next few decades, my portable computers have become ever more powerful and, thankfully, ever smaller and lighter. My current laptop is thousands of times faster than my first portable computer and it has thousands of times the storage and a gorgeous 15" color touch screen (vs. a 9" monochrome screen) - and it weighs a mere 3 pounds.

Since the late 2000s, I have worked almost exclusively on a laptop. I also have a desktop computer, both as a backup computer (in case the laptop has issues) and as a computer devoted to maintenance tasks like backing up servers in the early AM hours every night. Before that, I did most of my work on desktop computers, because they were faster and more powerful than laptops (at least any laptop within my price range). In that they have become rarities, I'll explain: a desktop computer is computer that sits on a desktop (or more often on the floor underneath) with cables to a separate monitor, keyboard and mouse. It has no battery, so it plugs into an outlet and it has a network cable connecting it to a router/hub (most desktops didn't have wireless networking - they don't move, so cabling them made sense).

Every afternoon before heading home from work, I copied the documents and programming projects I was working on from the desktop to whatever portable computer I used, both as an offsite backup and to allow me to continue my work at home. In the morning, I copied updated files from the portable computer back to the desktop computer and then worked all day on the desktop computer.

Previously, I mentioned 'servers'. As so many services have moved into the 'cloud', we have lost sight of the computers that power these services. All online services are powered by computers little different (in operation) than a laptop or desktop computer. These days, server computers are little more than a circuit board, containing a processor, memory, storage and networking ('blade' servers). Thousands of 'blades' are slotted into racks in massive data centers, powering services like Google, Facebook, and Amazon.

Before so much of our computing moved into the cloud, servers were more like desktop computers, but with the fastest processors and substantial memory and storage. When my company began hosting websites, we used a high-end desktop computer as a server. As hosting became a major part of our business, we bought two powerful server computers, each about 17" tall, 10" wide, and 23" deep. We hosted websites via one server and email, DNS and other services on the other, but for redundancy, either server could run everything (in case one server had issues). It was a great day when I got these servers up and running (because they could easily handle the load we placed on them), but it was an even better day when I moved all our services into the cloud, and I no longer had to manage the servers.

My computer bag has always contained a variety of wires and doodads that I might need while away from my office. From time to time, I browse the collection and purge the obsolete. My most recent foray into the bag led me to reminisce about some of the wires I used to carry. Until the mid-1990s, I carried an extension cord, because early portable computers had no batteries and the extension cord might be needed to reach an outlet. Later, when my computers gained batteries, they were power-hungry and had limited battery capacity, so even then I could not venture far from an electric outlet. For years, as I traveled through airports, I scanned the space just below the seating along the concourse, hoping to spot an electric outlet near a seat. In my earliest days carrying a computer, the only use for electric outlets along airport concourses was for maintenance people to plug in vacuum cleaners (there were no cellphones or tablets or games to charge), so there were very few outlets and not necessarily near any seating. If I saw an outlet near a seat, I rushed to take advantage of it.

For a few years, I carried a parallel printer cable, perhaps 15-20 feet long, a thick cable with big connectors on each end. As we didn't have email for sharing documents, we often printed things out. A printer cable was typically bolted to the back of a desktop computer and then traveled behind desks and file cabinets to a printer nearby, perhaps shared by several computers in the office. Before we had computer networks and networked printers, computers might share a printer using an 'A B C' switch, which allowed people turn a knob to specify which of several computers had control of the printer. When I was at a client's office, I might need to connect to their printers, and it was much easier to use my own printer cable than to use theirs, given that they were bolted to the back of a computer beneath a desk.

For over 20 years, I carried a 25-foot telephone wire and both an extender and a splitter. I used this wire to connect the modem in my computer to a phone jack (a plug-in receptacle in the wall) so that I could connect to the Internet (or earlier online services like Compuserve) via phone line. I used the extender to connect my telephone wire to another wire to reach even farther than 25 feet - because phone jacks were often far away from where it was convenient to use my computer. The splitter allowed me to connect two devices to a single phone line so that two computers could alternate access to the phone line or so I could connect both a computer and a fax machine. Starting in the mid-1990s, I carried a 10-foot network wire, to connect my computer to networks at clients' offices or to connect to the Internet from a hotel room. Network connections are an order of magnitude faster than phone connections, so this greatly improved what I could accomplish while on the road.

Through the years, I carried a video cable to connect my portable computer with its tiny monochrome display to a larger color monitor. I carried video adapters (to connect older computers to newer monitors or vice versa). I carried floppy disks, thumb drives, and external hard drives (for backup and file sharing). I carried extra batteries and battery extenders in my never-ending quest to keep my devices powered wherever I might go. I carried various tools to open up computers to add or swap peripherals (e.g. modems and network cards) and to make repairs to computers and peripherals - before devices became so small and intricate that I could no longer work on them.

So, in addition to carrying 30+ pounds of computer, I also carried a few pounds of wires, adapters and tools. Who needs a gym when you travel everywhere with such heavy baggage?

Phones & Mobile Devices

Before the 1980s, we used rotary dial phones connected by wires to 'jacks' in the wall. Shortly before my time, people might reference a location name (e.g. 'DA' for 'Davison') plus 5 digits (or just the 5 digits for calls within that local area). And before that, people tell stories of the 'party' line (particularly in rural areas) where several households shared a single phone line such that they could listen in on each others calls. But in the 1960s, my family had our own phone line and a 7 digit number - 782-7969.

It's comical watching young people attempt to use a rotary phone (check it out on YouTube!), but for the first 15+ years of my life, this is how we made a call. As we had no smartphones to remember or lookup phone numbers for us, we had to catalog and remember them ourselves. The phone company provided us with the 'phonebook', a massive list of all the people and businesses in your area and their

addresses and phone numbers (not much privacy back then). We all kept lists of important phone numbers and we remembered the phone numbers of family and friends who we called regularly.

When using a rotary dial, it's easy to make mistakes, like if your finger slips out of the hole before reaching the stop. It was annoying to make a mistake after a few digits and have to start over. There were many wrong numbers dialed in those days. In the 1970s, we moved to push-button phones, which made dialing numbers easier and more reliable (notice I wrote 'dialing' numbers, a language artifact still in use).

Phone numbers were connected to a place (like your house) and phones were connected to that place by a wire, often in a front hall or kitchen. There was one phone number for your house and all the people in it. When the phone rang, you ran to the phone and answered, very politely (as the call might be for Dad), something like "Hello, Hayhow residence", and the caller would then ask to speak with someone at the house. You then covered the mouthpiece with your hand (so the caller would not hear the shouting) and shouted "Dad, telephone", or in more civilized households you might tell the caller "just a minute" and you would quietly fetch Dad to the phone.

Much to the annoyance of their parents and siblings, teenagers monopolized the phone, chatting for hours with friends and sweethearts. As there was only the one phone number, no one could make a call while someone else was on the phone, and the house could not receive a call while another call was in progress. If you called a number already in use, you heard the 'beep-beep-beep' of the 'busy signal', and it might take several tries to get a ring instead of the busy signal. Many arguments were had about all facets of phone use/abuse.

Table-top phones (like the black one above) might have a lengthy wire between the phone and the wall, and teenagers might haul the phone into another room for some privacy. When my kids were teenagers, we had a wall-mount phone (like the beige one above) and I installed a long coiled wire so they could drag the handset into the family room for a more comfortable call. Often we had to dodge phone wires draped across walkways.

Many homes had multiple phones installed (all sharing the same phone line/number) so that you could answer or make a call from various places within the house. This also allowed more than one person to participate in a call, one on each phone. This also enabled invasions of privacy and buffoonery, when a kid might listen in on (or worse, comment on) a sibling's phone call.

Until the early 2000s, long-distance phone calls were expensive - while you could make local phone calls for free, you had to pay hefty per-minute fees for long-distance calls. This meant that we rarely called long-distance, and when we did, we kept conversations brief. It also led to tricks like hang-up messaging, whereby letting the phone ring twice, then hanging up before the call was answered, meant "I made it home", a message sent without incurring long-distance charges.

To reduce costs, we often used calling cards (sold by phone services other than your default local phone company) which offered lower-cost long-distance calling by allowing us to call a local phone number (free) and then make a long-distance call via the calling card. This entailed punching a lengthy series of numbers on the phone, which was particularly annoying when calling using a pay phone. For connecting my computer to online services, I could usually connect via a local phone number, which meant I could be online for a long time without having to pay for the call. This meant that when I traveled, I had to find local phone numbers to use in each place that I went.

Sometimes I needed to connect my computer to another computer via a long-distance call and sometimes I had to use a calling card to reduce the cost of the call. This entailed entering a long string of numbers interspersed with commas to indicate pauses. For example, from a hotel room, the string might include '9' to get an outside line, then a comma to wait for the dial tone for the outside line, then 7 digits for the local phone number for the calling card service, then 2 or 3 commas to wait for that call to be answered, then the 10-12 digit number on the calling card (to pay for the call), then another comma to wait for the dial tone for the long-distance call, and then finally the 10 digit long-distance phone number to call. It took some experimentation to get all this right because I might not have the right number of commas, and therefore numbers might be dialed before the system was ready.

Earlier, I mentioned the concept of the pay phone. Before cellphone use became widespread, pay phones were everywhere - on the wall at a convenience store or gas station, in a booth along a busy street, and collections of them in busy places like hotel lobbies and airports. To use the phone, you inserted coins (they could not accept either bills or credit cards) into the phone ($0.25 for a local call or quite a bit more for long distance), or you could use a calling card (by punching in all the needed numbers). Today, when a plane lands, everyone turns off 'airplane mode' and checks for calls or messages as the plane taxis to the gate. Before we had cellphones, we had to check in using pay phones in the airport terminal. Upon exiting the plane, there was as big a rush for the pay phones as for the bathrooms. I would call my office to check for messages and call home to update my status. Often, all the nearby phones might already be in use, so I had to wait for someone to end their conversation, which could be quite annoying if I felt their conversations were less important than mine.

Before people carried cellphones, you could only call them if you knew where they were, because phone numbers connected only to places, not to people. You might try to connect with someone at their office or at their home, but if no one answered the phone, or if it was busy (because they were on another call), your only choice was to try again later. Or if someone else answered the phone, but the person you wanted wasn't there, you might ask them to take a message or ask if there was another phone number to try (such as another office or a hotel). Starting in the 1980s, answering machines became affordable and ubiquitous, so even if no one was available to answer a phone, you could leave a voice message. As the machines became more advanced, and people became more technology-

savvy, you could call your own answering machine from afar and listen to your messages while away from home (otherwise, you could only hear your messages upon returning home). I know - this all sounds primitive, and it was.

In the late 1980s, I got my first mobile phone - a Motorola 'bag' phone, a purse-sized device that could be carried slung over a shoulder but which was usually used in a car. Cars used to have built-in ashtrays (yes, gross) and cigarette lighters, and the cigarette lighter could be used to power small electrical devices. I put the bag phone under the driver's seat and plugged it into the cigarette lighter for power and mounted an antenna wire to the back window with small suction cups. As of this writing, I still have that bag phone, though it has been non-functioning for many years.

Before I carried a cellphone, I wore a pager on my belt. A pager was a small device that had its own phone number, and you could call that number to leave a numeric message. Typically, you would send your own phone number as a request to have someone call you back. Pagers allowed you to contact someone regardless of where they were, which was a boon for both work and home life (we could give the pager number to a babysitter and enjoy a night out). Pagers were also the preferred method of contacting drug dealers to arrange a deal.

In the mid-1990s, I began to carry a Motorola MicroTac cellphone, a flip phone with a pull-up antenna. For the first time, people could call me without regard to where I was, and if I didn't answer, they could leave a voice message with the phone rather than at my home or office. This was incredibly liberating - I no longer had to check in at my office multiple times per day to get messages. It was also comforting for family - my wife could easily reach me wherever I might be. She quickly leveraged this into getting her own cellphone, arguing that if she had car trouble or was in an accident, she could call for help. Given that she was driving in busy metro Detroit with our three small children, I had no choice but to agree. Given that in the 1990s, cellphone calls were expensive (and charged by the minute), I got her to agree to limiting calls and chatting by cellphone.

It was not until the early-2000s that cellphone networks fully supported the sending of text messages, and years later before the cost-per-text became low enough that we didn't think before texting. Before smartphones, it was difficult (and slow) to send text messages because our phones did not offer a full keyboard for typing characters. Cellphones had physical buttons for the digits 1 to 10 plus # and * and a few other buttons to control the phone (e.g. 'send'). The alphabet was accessible via the number keys, three letters per key, such as 'ABC' on the 1 key. To type a 'C', you pressed the 1 key 3 times, once for 'A', twice for 'B', and a third time for 'C'. As cellphones became smarter, they offered 'T9' predictive text technology, which reduced the number of letter taps by guessing the word you wanted to type based on the letters already entered. Yes, given how we interact with our phones now, the old ways seem Neanderthal.

While I'm a whiz at typing on keyboards (having relied on them throughout my career), I'm an oaf at typing on smartphones. My fingers are too fat and I never learned the two-thumbed frenzy performed by youngsters who grew up with smartphones. I'm glad for the voice-to-text option on my phone and for the Windows Phone Link app that lets me use my computer keyboard to type text messages to send via my phone.

Through the years, cellphones got ever-smaller yet more powerful, batteries lasted ever-longer, and calling costs dropped ever-lower. But with the introduction of the iPhone in 2007, we went from carrying cellphones with a few simple apps to carrying Internet-enabled handheld computers. Smartphones changed everything, and the iPhone led the way. However, iPhones were much more expensive than

other devices, and the 'data plans' needed to access the Internet via iPhone were also very expensive. It was not until 2009 that I could justify getting an iPhone, and my rationale was not that I needed the device for my own use, but that I needed an iPhone to test how my website designs looked for iPhone users. Now it seems that everyone carries a smartphone - even elementary school children, who will grow up never having known a world without such devices. Much as my writing here, one day they will look back at our current miracles as archaic. I would love to know what new miracles will have replaced them.

Before I carried a smartphone, I carried a 'pocket secretary' (aka 'daytimer', aka 'planner'), a small notebook that fit in a suit jacket pocket. It was almost as essential to me then as your smartphone is to you now. It held my calendar, my contacts, and notes about everything. It had pockets into which I could stuff notes and business cards (we didn't exchange 'digits', we exchanged paper 'cards' imprinted with our contact information). It included reference materials like holidays and measurement conversions. Every January, I loaded a new, blank calendar for the coming year, and for a few months carried last year's calendar for reference. I still have a few calendars from the 1980s and 1990s and they are fascinating (at least to me).

In the early-2000s, many business people carried (and became obsessed with) BlackBerry devices, which included a built-in (but tiny) keyboard, and which popularized email messaging via mobile device. Perhaps because I no longer traveled much for work, I never felt the need to carry a BlackBerry - instead, I was (and still am) most comfortable handling my email via computer. From the 1990s until the adoption of smartphones, people used an array of devices with capabilities similar to the pocket secretary, perhaps most notably, the PalmPilot. I recall having a PalmPilot, but for me, it was not easier/better/faster than my pocket secretary and computer, so I never came to rely on it.

And before we carried smartphones, we used paper maps - there was no Google Maps, nor GPS navigation to guide us. In our own cars, we carried foldable paper maps in the glove box. At the airport car rental counter, the agent provided a local map and circled your destination, delineated the route, and offered suggestions on avoiding traffic and hazards. We studied the maps to retain the basic route concepts and then headed out. Fortunately for me, I have always had good instincts for directions and map reading, so I rarely had trouble finding my way.

We sometimes made mistakes or got lost, and the only alternative was to stop at a gas station to ask for help. A meme in those days (a time before we even recognized the concept of memes) was that real men don't stop to ask for directions. I guess that means we wandered around trying to find where we were on the map and how to get from there to our destination. I got lost a few times, including once, late at night in New Jersey, when I ended up in a neighborhood that I absolutely should not have been in. Every block further, the situation got scarier, until I saw a busy road off to the right, and went the wrong way down a one-way street to get there. I still don't like driving in New Jersey.

Another joy of paper maps was learning to read them. With smartphone and GPS navigation, you don't need to know much about where you are or where you're going - the system tells you all you need to know, including alternate routes when you encounter issues on your planned route. With paper maps, you need to solve your navigation issues on your own. My wife and I had many testy exchanges over

map reading when we needed a quick route change, and she struggled to find where we were on the map and what alternate routes were available. Eventually, she became a very good map reader.

Starting in the mid-1990s, we used Internet-based mapping services (e.g. MapQuest) to print out directions to a destination. Yes, we still used paper, but it was a nice improvement to have detailed directions, rather than just the paper map. A few years later, we used GPS navigation devices (e.g. Garmin), which provided interactive maps and turn-by-turn directions much like that provided by smartphones. We had to download maps and route data into the devices and deal with sometimes-spotty coverage from the GPS satellites, but these devices were a vast improvement over paper maps.

Many people still wear a wristwatch, and many now wear smartwatches, but soon after I started carrying a pager, I stopped wearing a watch. It's hard to say I'm into simplicity, but I only need one device to tell me the date and time. I have a small collection of old watches, from myself, my parents and my in-laws - fortunately, there are still collectors for such antiques.

Online Access

Until broadband and wireless technologies took over in the early 2000s, computers talked to other computers mostly via phone lines, using a device at each end called a modem (short for '**mo**dulator-**dem**odulator'). Some of the earliest modems linked to the phone line via an 'acoustic coupler' - you placed the handset of a phone into cushioned cups (with the cushioning as sound insulation), and your modem communicated with another computer/modem using audio signals, with beeps and a static-sounding language that was coded and decoded by the modems. I only used the acoustic coupler for a year or two - fortunately, improvements to modem technology soon allowed us to plug a phone wire directly into the modem.

My first modems operated at 300 baud, a tiny fraction of the speed of a modern Internet connection like the 50MB connection I have at home (which in 2023 is slow compared to the 100MB+ speeds offered elsewhere). In the early 1980s, I was happy to upgrade to a 1200 baud modem (a massive 13" Hayes Smartmodem card - installed inside a computer - which strangely I still have in my basement archive). From the mid-1980s to the mid-1990s, we upgraded to 2400 baud, then 9600, then 14.4K (where 'K' means 'thousand'), and finally 56K (which remains the top speed for 'phone' modems even today). Putting all this in perspective, it would take a fraction of a second to transfer a 1MB file (such as a small photo) via my 50MB connection. Using a 56K modem, this same transfer would take over 2 minutes. At 300 baud, it would take nearly 8 hours.

Of course, during that time we also used phone lines for phone calls, and this created conflicts between phone and computer use - if you were online, you couldn't receive a phone call and if anyone in a house was on the phone (most homes had only a single phone line), you couldn't go online. When my father built our new cottage at Viking Club (1996), he included double phone jacks in every room in the house, and outside on the porches, and even in the garage. With two phone lines, one of us could be online - without preventing incoming calls on the other phone line. Sometimes, two of us would be online at the same time, once again preventing incoming calls, but allowing two of us to get work done at once. It was a bit expensive (paying for two phone lines instead of just one), but it was worth it. Now, it seems

silly to have all those phone jacks because we all use cellphones and wireless Internet - we haven't had wired phone service for years.

Before we had broadband (high speed) Internet connections, we used modems to connect to online services like Compuserve and AOL (America Online), and later, the Internet. A generation of children, including my own, grew up hearing the bee-boops of their modem connecting to AOL, where they could hang out with their friends online long before the arrival of Facebook or TikTok.

dial-up-internet-sound.mp3

Soon after we moved into our house in Farmington Hills (1996), we added a second phone line for Internet access. In 2000, we acquired broadband Internet service via the newly available DSL service from Ameritech (a regional phone company reacquired into AT&T in 2003). DSL service was 'always on' and more than 25 times faster than the dialup access we had been using (1.5MB vs. 56KB). The DSL wire was connected to a wireless router which allowed multiple computers/devices to be online at the same time, and which allowed online access anywhere in the house - no wired connection needed. We take such fast/easy connectivity for granted today, anywhere we go, but in the year 2000, these were profound improvements.

At our remote Viking Club cabin, we relied on phone connectivity for several more years, until late 2009 when the club installed a point-to-point wireless system via a 100-foot tower (as of 2023, still on the SE corner of the property). For a time, the service was satisfactory, though somewhat unreliable - with numerous minor outages and slowdowns. Eventually, the provider (Cherry Capital Connection) proved unable or unwilling to provide service, so in 2015, we abandoned this approach. Fortunately, AT&T had installed a cell tower relatively close to Viking, so for a few years we got high-speed Internet via a 'hotspot' device, which connected to the cell tower much as a cellphone. However, the hotspot service was very expensive - we paid $50 for each 5GB of data we used. We mandated 'no silly cat videos' to discourage people from using data frivolously. A few years later, AT&T allowed us to use our cellphones as hotspots, so we no longer needed the hotspot device. Cellphone hotspots were limited to 10GB/month, but with multiple cellphones in the building, our Internet usage was much less limited. Finally, in 2021, Frontier Communications brought broadband (16MB DSL) to Viking, and for the first time, we had reasonably fast, reliable, unlimited Internet. This also allowed us to stream television via the Internet, which allowed us to abandon our DirecTV satellite TV service (see 'television' below). I offer thanks to the federal government for providing substantial funding for rural broadband initiatives - without them, we would still be hotspotting.

Television / Video

When I was a young child in the early 1960s, we watched TV in 'black and white'. While color TVs were available then, they were expensive, and because it was also expensive to create and broadcast color programming, only a few programs were broadcast in color. I remember going with my dad to the store to pick up our new color TV, in the mid-1960s. The new TV was enormous, with a heavy glass 'tube' (screen) and a vast array of bulky electronics stuffed inside a wooden box on legs - as if a piece of furniture. Stereo systems from back then were similar - a record player (phonograph), radio, and

speakers were often housed in wooden furniture the size of a dining room buffet. Our first experiences with color TV were like watching the Wizard of Oz, which starts in black and white and then explodes in color when Dorothy steps out of her fallen house in Oz. Soon, it became normal to watch TV in color and it seemed archaic to watch older black and white shows.

Today's electronic devices are powered by chips and components that are so inscrutably tiny that there is almost nothing we can do to make repairs on our own. The good news is that most of these devices and components are incredibly reliable. Before the 2000s, most of our devices were much less reliable, in large part because they relied on bulkier, less well-refined electronics. It was commonplace for a TV or computer screen to go 'wonky', with issues like static or scrolling or color glitches obscuring the view. It was also commonplace for us to smack the side of a TV or monitor to solve the issue, a practice which was surprisingly effective. With early TVs (especially color TVs), sometimes the issue would be more serious, like no picture at all (you could hear the audio, but see nothing on the screen), or video or audio glitches that weren't fixed via 'smack'. For such issues, it became necessary to make repairs.

Before semiconductor chips, electronics relied on 'vacuum tubes' to control the flow of electricity within a device. Vacuum tubes were notoriously unreliable, often burning out right at the start of your favorite show (which started a first round of cussing). Dad would then drag the TV out from the wall, remove the back panel, and investigate the clusters of wires, tubes, and other components inside (this started the second, and more profound, round of cussing). Sometimes you could tell which tube had expired by seeing soot on the inside of the glass, but usually, several tubes looked sooty - they ran very hot (and yes, they used a lot of electricity). So Dad would pull several tubes from the back of the TV, carefully so as not to burn his fingers, and alertly to remember which tube went where. Sometimes I went with him to the TV/appliance store where they had a device for testing tubes. The tubes varied in size and in the number and placement of connecting pins on the bottom. The testing station offered an array of plugs to accommodate any of the tubes you might bring in. One at a time, Dad plugged a tube into the corresponding plug and the machine would indicate whether the tube was good or bad. He then bought replacements for any bad tubes, and we returned home to plug them back into the TV, praying that this would fix the TV so that we could watch our shows. Sometimes the tube replacement didn't fix the problem (which inspired even more cussing) and either Dad would test/replace more tubes or on rare occasions, we would call for a TV repairman to come to the house to make repairs - because given the size/weight of the TV, there was no way we were hauling it anywhere for repairs.

It was not until the mid-1970s that my family lived where we could get 'cable' TV, via which we had a reliable signal and access to more than a few channels. Instead, we relied on antennas, ranging from small devices sitting atop the TV (for people in urban areas close to TV transmission towers) to roof or tower-mounted antennas (for people in more remote areas). In the 1960s, we lived on a farm a few miles outside the small town of Dowagiac, Michigan, many miles from any city big enough to have a TV transmission tower. So we had an antenna mounted on a tall metal tower (perhaps 30' tall), with a device inside the house to control the direction the antenna pointed, so that we could get the strongest signal for a given TV channel. We could get only 3 channels, and each time we changed the channel, we had to turn a dial to reposition the antenna for that channel, and it might take as long as 30 seconds for the small motor to turn the sizable antenna. Otherwise, the TV screen might be filled with 'snow' or the audio might be filled with static, such that it was unpleasant or impossible to watch. At our Viking Club cabin, we had a similar antenna (which barely brought in 2 channels) until we could install satellite TV (DirecTV) in the 1990s.

When a tower or roof-mounted antenna wasn't available, we relied on 'rabbit ears', two telescoping antenna rods extending up from the TV, which could be aimed to get the strongest signal for a channel. Weather sometimes compromised the signal, and if this prevented us from watching something like an important sporting event, we tried many approaches to improve the signal. I remember once struggling to watch a Detroit Red Wings playoff game while at an Indian Guide campout at a remote YMCA camp. We tried putting foil on the ends of the rabbit ears and we tried running wire from them to extend their reach. We even tried an often-used antenna hack by having someone stand next to the TV holding the antenna with one hand and reaching the other hand into the air. If this improved the signal, that

someone, usually (and unfairly) the youngest child in a group, would have to stand there so we could watch the game.

Having only 3 channels made TV watching very different from today. Newspapers published TV schedules for the 3 channels, and many people subscribed to TV Guide, a small weekly magazine listing all the upcoming programming. These schedules were important because we watched shows when they were on - not when we might want to see them. If you weren't sitting in front of your TV when the show was broadcast, you missed it and there was no recourse. The only way to find out what happened in a missed episode was to ask your friends. You could not record a show or stream it or pause it. Today we complain about commercials, but back then we needed them - it was the only way to get a bathroom break or to go get a snack, without missing some of the show.

The picture below shows a mid 1960s color TV, much like ours, suitable as a piece of furniture in our TV room/den (our 'living room' was reserved for higher purposes). We had no remote control, so we selected channels via a knob on the front of the TV. Sometimes we fought over that knob, when people wanted to watch different shows that were on at the same time. Sometimes we positioned a child (typically the youngest) at the knob to serve as our remote control - like when trying to watch two football games on New Year's Day. Sometimes we sat next to the knob so that we could quickly switch between shows when one channel went to commercial (that is, started showing ads, which might last 2-3 minutes).

From 1972 to 1974, my family lived in Brussels, Belgium (a story shared elsewhere). We had only a black-and-white TV and could access only a French-language channel and a Flemish-language channel, the two languages spoken in Belgium. We preferred the Flemish (Dutch-like) channel because it often displayed English subtitles, which allowed us to take in the dialogue. Due to lack of programming, we didn't watch much TV there.

By 1975, we lived in East Lansing, Michigan, and enjoyed our first experience with cable TV and our first TV remote. Our cable service came with a box, wired to the TV, with a wire long enough that you could sit across the room and control the TV, which was quite a luxury. The box had a dozen buttons and a 3-position switch to offer a potential of more than 30 channels, although only a dozen included programming of any usefulness. But the signal was clear (via wired service instead of antennas) and we had access to programming far beyond anything we had previously experienced.

The rapid development of semiconductor technologies in the 1970s led not only to computer technologies but also to incredible advances in both television and music, including the Video Cassette Recorder (VCR), video discs, handheld video cameras, and CD players. When these technologies were first introduced, they were prohibitively expensive - VCRs cost over $1000 ($5000+ in 2023

dollars) for the first few years after they were introduced (in the mid-1970s), but fell below $400 by the mid-1980s and below $200 by the mid-1990s.

We didn't get our first VCR until the late-1980s when the price became somewhat affordable (at that time, we were still young and poor). For the first time, we could record and pause TV shows, and we could buy and rent movies and shows to watch at home. Our oldest son Brian learned the rewind and play buttons as a two-year-old so that he could watch his Raffi video over and over (Raffi performed much-loved songs for children).

VCRs gained the ability to record a specified show at a specified date and time, but it was so difficult to program the VCR that a common joke of the day was "we can't even set the time", our VCRs blinking 12:00 instead of the current time. Instead, we just waited until a show was about to start and pressed 'record'.

A few years later, we acquired a handheld video camera (Sony Handycam), I'm sure, as a gift from my parents so we could record and share videos of their grandchildren. Before that, my father often traveled with his shoulder-mount video camera which recorded onto full-size VCR tapes. In contrast, the Handycam was much smaller and recorded onto smaller tapes. As smartphones with video recording capability would not show up until the late 2000s, the Handycam allowed us to capture the highlights of our children's early years.

Much like the VCR before it, when introduced in the late 1990s, the Digital Video Disc (DVD) player was prohibitively expensive. DVDs had much higher video quality than VCR tapes and they did not suffer the reliability issues we had with video tapes (which would regularly get snagged inside the machine, ruining the tape). We acquired our first DVD player in the early-2000s when the prices became affordable. The 1990s and 2000s were the glory years for VCR and DVD movie rentals, first with the ubiquitous Blockbuster and various other movie rental stores and then with Netflix offering movie rentals by mail (that's right, they mailed us the movies we wanted and after watching them, we mailed them back). By the 2010s, we could rent movies via Internet streaming or via cable or satellite TV services, so our use of tapes and discs dwindled to nothing.

Earlier, I described how our 1960s TV was a heavy glass tube with a mass of electronics housed in a wooden cabinet on legs. However, the viewable screen was probably no bigger than 23" so yes, we had to sit fairly close to the screen to get a good view. By the early 1980s, screens had much better video quality, but typical screen sizes had not increased by much - I think ours was 27". From the late 1980s through the 1990s, screen sizes steadily increased, but still using the same tube technology (CRT). The largest tube TV we owned was a 36" that was so bulky and heavy that it was challenging for two people to move it. In the early-2000s, flat panel TVs were introduced but as with all new technologies, they were prohibitively expensive. By the mid-2000s, prices had dropped to somewhat reasonable levels, but still more than $2000 for a TV. It was not until 2010 that we acquired our first flat panel TV, a 50" plasma TV. The larger viewing area and wider screen ratio improved our TV-watching experience dramatically, especially when watching movies. As of 2023, we still have that TV, and it still works well as a secondary TV, though it is dwarfed in screen size and quality by our 65" primary TV (itself dwarfed

by the 85" and larger TVs available today). Due to ever-improving technology, our 65" TV is half the thickness and weight of the older 50" TV.

In 2023 we no longer rely on antennas or cable for TV programming - everything we watch arrives via Internet. We subscribe to a streaming service (DirecTV Stream) for traditional TV channels, live sports, etc. We subscribe to multiple content streaming services (Netflix, Hulu, Apple TV) for on-demand movies and shows. While we have a wealth of content sources to watch, we sometimes struggle to find something we want to watch, and we almost always struggle to figure out which service hosts a show we're looking for. We hope that one day this will be much easier.

Music

When I was a child, we listened to music on the 'radio', a device that received signals transmitted from radio towers near and far. Depending on the distance and the weather, we might get a clear signal or the music might be sullied by static. From our house in the countryside of southwestern Michigan, we could (on a good day) listen to WLS in Chicago, which played current hits (1960s) rather than the less hip offerings of the local radio stations. WLS provided an AM signal, which traveled farther than higher-quality FM signals, so we traded audio quality for the better playlists offered by WLS.

And, as our parents before us, we listened to 'records', 12-inch black vinyl record albums played via a 'record player' (aka turntable). Record players had a small peg/spindle for full albums and a '45 adapter' which fit over the small peg to accommodate the larger center hole of the (ironically) smaller 7-inch records (called 45s) which held a single song on one side and another song (the 'B side') on the other. By the late 1960s, record players included a spindle that allowed us to stack up to 6 albums which were released one by one (with a distinctive smack) to let us listen to 6 albums of music without intervention. Sadly, this allowed listening only in the order the albums were stacked and in the order the songs were arranged on an album side. As albums are two-sided, this meant skipping the other side of the album (until later when you flipped the album).

In the early-1970s, our music listening was upgraded via new audio 'tape' technologies. First, we enjoyed the 8-track tape, with flimsy ribbon tape housed in a plastic case the size of a sandwich, and which played continuously in a loop. The audio quality was poor, and the tape often became snagged within the player, but most importantly, unlike record players, 8-track players could be installed in cars, allowing us to choose what we listened to rather than listening only to what was offered via the car radio.

By the mid-1970s, we enjoyed a significant upgrade with cassette tapes, smaller-sized like a deck of cards, with better audio quality, and most importantly, the ability to record music rather than just play it. When I was 14, I bought a stereo (our name for 'music player') that combined a radio, record player, and cassette tape recorder, which allowed me to record from either the radio or record player onto a tape. I could tape music from the radio (for free rather than paying to buy a record/album) and I could create 'mix tapes' by recording selected songs from various albums onto a tape. Many of us fancied ourselves as gifted DJs, creating music mixes for parties, road trips and, of course, wooing women.

At the peak of my interest in current music, I had more than 300 record albums and dozens of cassette tapes. Much as book lovers proudly display their libraries, we displayed our album collections, and when visiting friends, we often browed their collections as a way of getting to know them, or to discover new music. Later on, when CDs rendered record albums obsolete, many of us abandoned our record collections because they were large and difficult to move, and because our wives did not share our nostalgia for them.

As with computers and TVs, the rapid development of semiconductor technologies in the 1970s led to great advances in stereo equipment. In 1977, I bought a mid-level receiver (a combination of a radio tuner and an amplifier), a turntable, and a pair of speakers (Ohm C2) which were about two feet tall and a foot wide. With this gear, I could really make some noise. Most of my friends had similar equipment, and a few had more expensive gear, with speakers reaching 3 or 4 feet tall. Early in my marriage (the late 1980s), I remember having a 3-foot stack of matching stereo equipment in a glass/plastic cabinet in the living room (I think the bottom shelf stored record albums). Within a few years, our interest in playing music faded as raising children became our focus and we got rid of the fancy stereo equipment.

By the late 1980s, most music was distributed via Compact Disc (CD) rather than records or tapes. CDs offered much higher audio quality and the ability to skip directly to a song (not as easily done with records or tape). In the late 1990s, CD-R and CD-RW technologies allowed us to copy songs from our computers onto CDs. We could copy songs from CDs (and other sources) to our computers, create a playlist, and then 'burn' the music to a CD, much like the mix tapes we created before CDs.

And then, in 1999, Napster changed everything. Napster (and a variety of similar services) allowed us to download any songs we wanted for free (anonymously from other peoples' computers on the Internet). While not technically legal, there were no barriers to prevent such music sharing, and for several years, everyone had free access to all the music they might want. I confess that I collected all the classic music from my youth, a collection I still keep (several thousand songs). Within a few years, legal pressures forced these services to shut down.

Much as video streaming killed the videotape/DVD marketplace, by the mid-2000s, music streaming services like Pandora, Spotify, and Alexa (Amazon Music) supplanted the CD marketplace. While in the 1970s and 1980s, we had racks of stereo equipment and massive speakers, in the 2020s, most music is played via smartphone and relatively tiny wireless (Bluetooth) speakers.

Throughout my lifetime, we've had ever-smaller and better portable devices for playing music, but I've used almost none of them. In the 1960s, it was transistor radios; in the 1970s, it was boom boxes, in the 1980s it was the Sony Walkman; in the 1990s, it was MP3 players; and in the 2000s, it was the Apple iPod. I'll listen to music at home or in the car, but when I'm walking, hiking, or biking, I prefer to listen to the sounds around me, so I've almost never used portable music players.

Video Games

When I was a teenager (in the mid 1970s in East Lansing, Michigan), my friends and I often visited the local pinball parlors (aka arcade) to play pinball. At our homes, we played board games, ping-pong, and

pool, but we had no video games because they didn't yet exist. At the arcade, we played older pinball games from the 1960s, which were mostly mechanical and newer 1970s games with more advanced electronics. Games cost a quarter to play, and most of them offered methods of winning a free game (by scoring enough points or targeting a feature within the game). In that we were usually short on money, it was crucial for us to get good enough at the games to consistently win free games - often I could play for an hour or more on a single quarter.

Video games entered my life with the invention of Pong, a primitive version of ping-pong, which we first played at the pinball arcade as a stand-up game with a TV screen and two knobs which twisted to control the on-screen paddles. A home version soon followed (1975) via a console/box that connected to the antenna inputs on the back of a TV. Pong was (and still is) a boring game, much less interesting than ping pong, but we were mesmerized by the technology - by the concept that we could play games on our TVs.

In the late 1970s, pinball parlors began to offer a few video games, more each year as new games came out, taking the place of pinball machines, until we had transitioned from pinball parlors to video game arcades. At first, there were only a few games, so most of us became quite good at them (Space Invaders, Breakout, Asteroids). By the early 1980s, there were many different games, and we began to specialize on games we were good at, in hopes of setting high scores on each game machine.

The first home video game consoles were introduced in the late 1970s. Many people played games on the Atari console (Atari having been one of the pioneers of video game technology), but I picked the Magnavox Odyssey 2 (knowing me, probably because it had better specs or a lower price tag). The games were rudimentary, and the graphics were primitive, but much improved from the days of Pong. I kept the console and games for many years, allowing my kids to marvel at the crude games - compared to the far more advanced Nintendo and X-Box games they played (now obsolescent themselves).

As I finished college and started work, I had little time for games, so I did not keep up with the progression of video game consoles and games. I never learned the intricacies of multi-button video game controllers, and therefore when my kids tried to teach me their games, I failed horribly.

As a break from work, I have always played relatively simple PC-based and web-based games like Tetris or snake. And as I age, I play a variety of word games like Wordle, hoping to keep my mind sharp. But more sophisticated video games are far in my past.

Printing / Faxing

When I was in school, we completed most of our assignments in hand-written form. For more serious works like reports, essays and term papers, we used the typewriter, a mechanical device with a keyboard where each key press bashes a character through an ink ribbon onto the paper. As with any typing, it was easy to make minor mistakes, but unlike using a computer, there was no backspace function - the mistyped character was already printed on the paper. Instead, we might just 'xxx' over the mistyped characters, or for better results, we might use an eraser (like on a pencil) or 'white out', a quick-drying white liquid that we painted over the mistake.

As a freshman in high school, I took a typing class, both to improve my keyboard skills and because the class was mostly attended by girls (as typing was a required skill for secretarial work that was a sexist and likely career path for women in the 1970s). In this class, we used the IBM Selectric electric typewriters, via which typing was faster and less tiring, because electric typewriters move the print head electrically rather than by the force of your fingers pressing down on the keys. The Selectrics also offered a backspace function which repositioned the print head back one space and then retyped the bad character through a 'white out' tape to (mostly) erase it. For me, gaining proficiency in typing would pay great dividends in my future career in computers and business.

Through the early years of my computer experience, we did most of our printing via dot-matrix printers, which bash tiny pins through an ink ribbon to form letters. These printers were loud and produced poor-quality output, but they were inexpensive and relatively fast - much faster than the typewriter-like printers that preceded them. For high-quality output, we used relatively slow daisy-wheel printers, which use a wheel containing a set of printable characters each like a petal on a daisy - the printer spins the wheel to position each character and bashes the character through an ink ribbon onto the paper. By the late 1980s, with the advent of the Hewlett-Packard LaserJet, we switched to laser printers, which were much quieter and faster.

These early printers could print only in black and white. The LaserJet enabled us to print shades of gray, but it was not until color inkjet printers were introduced in the late 1980s that we could print in color. But because early color inkjets offered relatively poor print quality and expensive ink cartridges, we rarely printed n color. In the 2000s, with the advent of color laser printers and better/cheaper inkjet printers, we finally began to regularly print in color.

Given that no one uses fax machines any longer, I'll explain what one is. A fax machine uses an optical scanner to create digital images of printed pages and then sends the images via telephone line to another fax machine which receives and prints out the images. We used fax machines to send documents and forms instantly, a great improvement over sending such documents by mail or Fedex. Back in the 1980s, we sent important documents across the world via Fedex ("When it absolutely, positively has to be there overnight"). By the 1990s, fax machines became affordable and ubiquitous and we could even send and receive faxes directly via our computers. By 2010, I had quit sending faxes, and only received occasional spam faxes - typically from travel agencies touting discounted trips to tropical locations.

Monitors

In college and early in my career, I used computer terminals that displayed green characters on a black screen, with a maximum display size of 80 columns by 24 rows. My early portable computers included similar displays, though smaller at 9" compared to 12" for the terminals. Often, we plugged our portable computers into large monitors, both for more on-screen workspace and also for color. Early color monitors like the NEC Multisync weighed over 20 pounds, so they did not travel with us. It was not until the late 1990s that laptops with color displays became common (and affordable).

In the mid-1980s, even with an external monitor, the screen resolution (EGA) was limited to 640 x 350 x 16 - that is 640 pixels wide by 350 pixels tall with up to 16 colors (from a palette of 64 colors). In the late 1980s, we upgraded to VGA, with 640 x 480 x 16 (with 16 colors from a palette of 256 colors). By comparison, the 15" laptop I'm writing this on (from 2023) offers 1920 x 1080 resolution and a relatively unlimited color palette. The low pixel resolution and color palette are why early computer games look so simplistic. The equipment we used was incapable of displaying anything more realistic.

Storage

First, I'll explain the abbreviations. KB stands for kilobyte, which is 1,000 characters (actually 1024 as computer memory and storage are based on binary numbers). MB stands for megabyte, which is 1,000,000 characters, and GB stands for gigabyte, which is 1,000,000,000 characters. So 1MB is a thousand times bigger than 1KB, and 1GB is a thousand times bigger than 1MB and a million times bigger than 1KB.

My first personal computer (1983) was the Victor 9000 (see chapter 1), which included two 5 1/4" floppy disk drives, each capable of storing 600KB of data. This was far superior to the original IBM PC, which had only a single floppy disk drive capable of storing 320KB of data. To put this in perspective, this article has now reached beyond 60,000 characters (60KB), so a mere 5 copies of this article would fit on the original IBM PC floppy disk. Given their small storage capacity, we needed to use many floppy disks, and we often needed to swap them in and out of the computer.

By 1983, the next model of the IBM PC (XT), included a 10MB hard drive, so we could store the equivalent of more than 30 floppy disks without needing to swap disks. A year later, the next model (AT) included a 20MB hard drive. Every year since then, it seems, hard drives get smaller and faster, and they hold more data. In the late 1980s, my company needed more storage so that we could share data across our network. As I recall, we bought a storage system that held 100MB at a cost of more than $1000. In comparison, today (2023), I can buy a 1TB (1 terabyte = 1000GB) drive for $50. So, in 1980-something, we got 100KB per dollar spent and in 2023, we get 20GB per dollar (a factor of 200,000 more storage per dollar).

For decades, we relied on the storage in our own computers. Larger businesses installed servers to provide shared storage for all the computers on their networks. As Internet speeds went up, it became possible to use offsite storage and we witnessed the birth of 'the cloud', whereby we use storage in massive data centers operated by companies like Apple, IBM, Amazon and Google.

Career - Appendix B - Incite / Wincite

Competitive Intelligence & Knowledge Management

Wincite was one of many products in a genre called Knowledge Management (KM), a term coined in the late 1980s, and a software category that surged in the 1990s. The basic premise of KM is to catalog knowledge stored throughout an organization and make it accessible to everyone in the organization. Particularly in large and diverse organizations, knowledge is siloed, stuck within a single subsidiary or department. For example, R&D staff may know that a competitor is building a product based on a new technology, but sales staff may be unaware.

Most of the early KM solutions were 'document management' systems, within which mountains of documents were consolidated within a single document database, which could be searched using keywords. Wincite was unique in that it was a structured database, with its structure based on the questions an organization needed to answer, about competitors' products, facilities, capabilities, etc. Wincite consultants interviewed key marketing and technical staff to document these questions, and then build a custom Wincite database to hold the answers.

One essential feature of Wincite was its ability to record metadata, information about the source and meaning of information recorded in the database. This enabled people in disparate parts of an organization to more easily collaborate in the research and analysis of competitor information. For example, R&D staff might encounter information that had been acquired by sales staff, and the metadata would enable them to contact each other.

Two new technologies led to Wincite's decline. First came Lotus Notes, a 'groupware' application, that offered organizations shared discussion, email and document databases. While Wincite required dedicated and well-trained staff to acquire and manage structured information, Lotus Notes connected everyone within an organization for less rigorous, but more widespread, information sharing. As a communications platform, Lotus Notes offered benefits far beyond what Wincite offered.

Then came the Internet, and the wealth of communication and application technologies it enabled. While Wincite Systems developed an Internet-enabled version of the software, Wincite sales declined through the late 1990s. In 2023, it appears the company is no longer in business.

Portable Library

To facilitate the sharing of reports and documents, we created an add-on module called the Portable Library. Because Wincite preceded widespread Internet and email usage, the information within a Wincite installation could not be easily shared. With the Portable Library, analysts could create a collection of reports and documents and 'burn' them onto a CD along with the Portable Library software. They could distribute these CDs to people throughout a company for simple sharing of competitor intelligence.

ForeCite

To facilitate browsing numeric information like sales and financial information, we created an add-on module called ForeCite, which worked much like the Traverser software we created at Price Waterhouse to browse data within the FM80 system. You could select Wincite's 'dimensions' (e.g. Company, Location, Product, Market, Year) and 'EEIs' (e.g. Sales, Cost of Goods, Net Income) to produce a spreadsheet with dimensions/EEIs assigned to the rows and columns for specified entities and time periods. You could then 'drill down' into any of the numbers to produce a new spreadsheet. For example, you might see a spike in Cost of Goods last year and drill-down to see Cost of Goods by division to research the source of the spike.

Incite/Wincite Customers (incomplete but representative list)

3M Automotive - Detroit, MI (automotive supplier)
Aetna - Hartford, CN (insurance)
Allied Signal - Morristown, NJ (automotive, aerospace)
Ball Corporation - Muncie, IN (canning products)
Bell Atlantic - Philadelphia, PA (telecommunications)
ChemWaste (Waste Management) - Oakbrook, IL (hazardous waste disposal)
Ciba-Geigy - Toms River, NJ (pharmaceuticals)
Dow Chemical - Midland, MI (chemicals)
Eaton Corporation - Beachwood, OH (automotive supplier)
Educational Testing Systems (ETS) - Princeton, NJ (SAT and other testing services)
Electronic Data Systems (EDS) - Plano, TX (computer services)
Fiserv - Brookfield, WI (financial services technology)
Ford - multiple divisions
G.D. Searle - Skokie, IL (pharmaceuticals, Nutrasweet)
GATX - Chicago, IL (railcar leasing)
GM - multiple divisions
Himont - Wilmington, DE (plastics)
Johnson & Johnson - New Jersey (personal products)
Kelly Services - Troy, MI (employment services)
Lederle (American Cyanamid) - Wayne, NJ (pharmaceuticals)
Mass Mutual - Springfield, MA (insurance)
Merillat (Masco) - Adrian, MI (cabinets)
National Semiconductor - Santa Clara, CA (computer components)
Owens Corning Fiberglass - Toledo, OH (building materials)
PPG - Pittsburgh, PA (paint, glass, coatings)
SC Johnson - Racine, WI (household cleaning supplies)
Shell Chemical - Akron, OH (plastics)
SmithKline Beecham - Philadelphia, PA (pharmaceuticals)
Timken - Canton, OH (steel/bearings)
TRW Steering & Suspension - Detroit, MI (automotive supplier)
Uniban - Medellin, Columbia (bananas and other agricultural products)

Unilever - London, England (consumer goods)
Whirlpool - Benton Harbor, MI & Comerio, Italy (appliances)
WR Grace - Boca Raton, FL (chemicals)

Career - Appendix C - Direct Knowledge Portfolio

Apologies

You might find the next few sections boring. While these projects have been fascinating and lucrative for me, and they represent thousands of hours of my work, I recognize that most people will get little entertainment from reading the details of my most important projects. Feel free to skim or skip ahead as needed.

Web Applications

In the early days of the web, a 'website' was a website and 'software' was delivered via CD/DVD and installed on your computer. Over time, developers (like me) created software that was deployed via the web browser - we call these 'web applications'. Web applications have tremendous advantages over traditional software. First, users don't have to install any software and a solution can be widely deployed and easily supported. Second, software updates and bug fixes can be implemented instantly, by updating the software on the server (rather than installing updates on users' computers).

I built my first significant web application in 1999, an assessment management system for Home Health Care agencies to facilitate Medicare reimbursement for their services. In 2001, I built a web application for an employment broker to help their US customers get H2B visas for Mexican workers, and to register thousands of workers in Mexico. In 2004, I built a web application for an insurance company to allow their customers (shopping mall operators) to manage information and documents about their tenants and vendors (as required by their insurance policies).

In 2006, I started work with StraightLine, a financial services company helping thousands of people manage the investments within their 401(k) and 403(b) accounts. I built a customer portal to allow StraightLine customers to view their financial position and to facilitate communications between customers and StraightLine staff. This web application allowed StraightLine to import financial data from various providers (e.g. Schwab, Fidelity, etc.), a process which is wildly complicated due to vagaries of the data formats between various providers. In 2007, I added a billing system to generate billing amounts based on customers' holdings and contract formulae, to produce print-ready invoices, to process credit card transactions, and to apply payments as they come in.

All of this was complicated by StraightLine's insistence that the application data must be stored via Salesforce.com (the #1 provider of Customer Relationship Management [CRM] solutions), which StraightLine uses to securely manage customer information. In 2011, I rebuilt the application within the ActiveK environment, primarily to allow StraightLine to manage the content presented by the portal, but also for security and performance improvements. Finally, in 2023, after over three years of

development, StraightLine replaced this application with a new application built by one of the leading Salesforce.com consulting firms. Apparently, my work is difficult to replace.

In 2008, I built a prototype web application for OfferSubmission.com, a new venture created to help banks sell homes they had foreclosed on (a nasty side-effect of the 2008 financial crisis). Via the application, banks could offer homes for sale, and real estate agents could submit offers to purchase the homes, and bank staff could review and manage the offers. To date, banks have sold several thousand homes via OfferSubmission (although sales have declined dramatically in the past few years), and OfferSubmission became my largest customer. Through the years, I built similar applications for them, one to help banks sell homes via short sale (a venture that didn't produce much success), and another (SellerNation.com) to allow agents to list/sell homes directly (also not yet producing much success).

My relationship with OfferSubmission was sometimes testy. They weren't very technical and they tired of answering the questions I needed answered to build them the elaborate system they wanted. Within the first year, as the project became profitable, they hired Ray Stefanski as their technical intermediary. Our relationship was bumpy at the start, but we soon developed a good partnership (even a friendship) in designing new capabilities and solving problems. In 2012, OfferSubmission decided that I was 'too expensive', and they contracted with a firm in India (Clarion) to develop new 'dashboard' and 'reporting' capabilities. Clarion's hourly rates were a fraction of mine, but they struggled to understand the specifications and OfferSubmission struggled with both time lag (make a request today, get the answer tomorrow) and language barriers. Several months later, Clarion finally delivered the new software, which was buggy, awkwardly constructed, and difficult to support. A year later, I convinced Ray that their reporting functionality was poorly conceived. All the reports were 'detail' reports, that generated many pages of results. Instead, I convinced him they needed a 'top-down' approach (much like the Traverser system I worked on at Price Waterhouse). In a few days, I built a mock-up of the system, via which you can view aggregated statistics, and then 'drill-down' on any of these values to view the underlying details. OfferSubmission realized they had paid Clarion to build the wrong thing, and that my understanding of their business made my higher hourly rate worthwhile.

Another long-term customer has been Rex Materials Group (RMG). Starting in 2005, I worked with RMG to develop and deploy a new corporate identity and build a new website. I admit that 'corporate identity' was not a service that I offered, nor was it within my skill set. However, it was the entry point for this opportunity, and I needed the business, and I subcontracted the design work with my long-time associate Chris Van Wyck of Ciesa Design (I first encountered Chris when he designed the print catalogs for my parents' Michigania store). For over 10 years, Chris and I (and later on Becky Faber, another designer I worked with) designed all RMG's marketing materials, including logos, business cards, stationery, brochures, product data sheets, advertising, banners and more. This (rightfully) was the only customer for whom I provided such services, and I was glad when they took the job in-house in 2017. In 2023, despite RMG having been acquired, not once, but twice, I still manage their website (but probably not for long).

In 2010, I built a custom web application for Weaver Media (WM), a company that installs advertising for car racing events (e.g. Nascar) and sports arenas. WM had an existing application (WMOnline) that

was poorly designed and even more poorly constructed. I built a replacement application (within the ActiveK environment), with a much better architecture and features far beyond the existing application - and I charged them less than they had paid for the existing application. WMOnline allows 'sponsors' (e.g. Home Depot) to place orders for advertising to be installed at 'facilities' (e.g. Daytona), with the materials to be supplied by 'vendors', and the entire process managed by WM. Each of these constituents can log in to the system to manage orders applicable to them and their role in the process. For each facility, WM maps all the locations (including specifications and photos) which are available to display advertising (e.g. 'the middle panel of the wall in the 3rd turn on the racetrack' or 'the support column in section 3a of the basketball arena'). A sponsor can order advertising for any of these locations, if they are available for a requested date range (because Home Depot may want to advertise for the Daytona 500, but not for a lesser race). Sponsors upload the artwork for their advertising, WM designs the advertising to meet the specifications, vendors produce the materials and ship them to the facility, and WM installs the advertising as needed. WMOnline is integrated with QuickBooks so WM can manage invoicing and payment. It is a complex business, and therefore a complex system. In 2023, WM continues to rely on this system.

Content Management System (CMS) Websites

Of the dozens of CMS websites that I have built, several stand out as noteworthy.

I first built a 'static HTML' website for Viking Club in 1998. In 2008, I built a replacement website using ActiveK, with the goal of preserving the entire history of the club, both past and ongoing, in website form. As of 2023, I have scanned and added several hundred documents to the website, dating back to the club's founding in 1936. My work is ongoing.

In 2000, I built a 'static HTML' website for our church, the Universalist Unitarian Church of Farmington (UUCF). In 2009, I built a replacement website using ActiveK, and I used this site as a 'guinea pig' for several extensions of the ActiveK software. First was the concept of 'public' vs. 'private' sections of the website, with the 'public' website available to everyone (to promote the church), and the 'private' website requiring login for access to member information including directories, bulk email, and committee activity. I also added 'blog' functionality to allow our minister to manage a blog within the website, audio and video players to present sermons, 'RSS' functionality to allow visitors to subscribe to updates (e.g. weekly sermons), and many Search Engine Optimization (SEO) techniques. In 2013, others at the church felt that the website needed to be more SEO-friendly, more mobile-friendly, more analytics-friendly, and built using one of the more popular CMS (WordPress) known to more church members. I still hold a grudge, because this group took over the website and rebuilt it without the participation of all the constituents, and because despite my having managed the website for over 10 years, I received little in the way of a thank you, and because to this day, the new website is unattractive and less useful (to members) than what I created. So there.

Universalist Unitarian Church
of Farmington, Michigan

Go: About UUs | Member Info

Thanks for visiting our website. We welcome the opportunity to introduce you to the UUCF family. As you browse our website we hope you will get a flavor of the energized congregation that we are.

Our meetinghouse provides a link between the rural congregation that originated in downtown Farmington as a liberal religious voice of reason over 150 years ago and the suburban congregation we are today. We remain today dedicated to the ideal of providing our membership the opportunity to explore and develop a deeper understanding of the life process and our place in it while maintaining the familial and supportive ties of yesteryear.

- Home
- About UUCF
- About Unitarian Universalism
- Religious Education
- Weddings
- Member Info
- Contact UUCF

Sunday Services:	9AM & 11AM (see calendar)
Minister:	Alexander Riegel
Director of Religious Education:	Kathy Silver
Music Director:	Diana Munch

NEW! Click >here to fill out the 2003 UUCF Congregational Survey.

NEW! Click >here to listen to audio versions of recent sermons

Copyright © 2000 by the Universalist Unitarian Church of Farmington.
25301 Halsted Road, Farmington Hills, MI 48335
phone: (248) 478-7272, email: info@uufarmington.org

Unitarian Universalist Association of Congregations
"Affirming the Inherent Worth and Dignity of Every Person"

Login | Location | Calendar | Contact Us

UNIVERSALIST UNITARIAN CHURCH OF FARMINGTON
Many paths One heart

Content
Search:
GO
advanced search

- Home (3)
- First Time Visitors
- About UUCF
- About Unitarian Universalism
- Sermons
- Church Community
- Religious Education
- Lifelong Learning (2)
- Social Justice
- Weddings & Events (1)
- Contact Us
- Calendar

A A A

Welcome

You're on your own spiritual path. We all are. Are you a seeker? Looking for a welcoming place to search your own truth? Just curious? Check us out. The Universalist Unitarian Church of Farmington welcomes everyone. UUCF. Many paths, one heart. more

Order of Service

Orders of Service are provided in Adobe Acrobat (PDF) format. If you already have the Adobe Acrobat Reader software installed on your computer, just click the links below to view each issue. If you don't yet have the Adobe Acrobat Reader software, just click on the 'Get Acrobat Reader' link and follow the instructions to

QUICK INFO

Sunday Services: 10 30
Phone: 248 478 7272
Map/Directions: Click Here

COMING EVENTS

No upcoming events found. Click here to view prior events.

Click for Farmington, Michigan Forecast

We Stand on the Side of Love
WILL YOU?

In 2009, I worked with Pace & Partners and Ciesa Design to develop new websites for a few satellite schools of Ferris State University (FSU). Pace provided marketing services, Ciesa provided design, and DirectK provided the website/CMS. This was one of the most rigorous proposal processes I have ever been part of, starting with a formal RFP and proposal, then in-person presentations, then vetting by FSU website and technical staff. It was also the most team-oriented (and therefore arduous) website projects I have worked on. It took over 6 months to design and build the sites, which included a wealth of information for students, prospective students, and faculty. The project was challenging on many levels. For such a prestigious customer, and as a subcontractor of Pace, I had to provide rock-solid software. Given that FSU hosted the site themselves, I had to provide support remotely. One of FSU's requirements was that we had to use Microsoft SQL Server as the database, which I had never used. This project pushed me, but in the end, we delivered a great solution for FSU. Unfortunately, within a couple of years, FSU standardized on a single CMS for all of their websites, and they abandoned the websites we had created for them. Putting it in perspective, they replaced the ActiveK software, developed by a single person, with Omni (now Modern Campus), a CMS now used by 1800+ institutions. When I asked my FSU contact to compare ActiveK and Omni, the only significant difference was that Omni provided user access and workflow rules, functionality needed for a large multi-user environment like FSU.

Apparently, I was very busy in 2009, because I also built a new website for Edward Rose & Sons (ERS), an apartment complex builder/management company. The proposal process was quite different than with FSU. ERS had been burned by an out-of-state website developer (who offered slow/poor support), and they wanted someone who could visit their office as needed. Because their office was less than a mile from my house (and because I was superbly qualified), I got the job. This project was also quite demanding. As with FSU, ERS planned to host the website on their own servers. They had an existing CMS-driven website, which meant that all the website content would evolve while the new website was under construction. I had to build an import function to grab the content of their existing website pages, reformat it for the new website, and then load it into the ActiveK system shortly before the new website went 'live'. I built a custom Property Management System to allow them to manage all the information and photos for each of their apartment communities and the various units within. I built search functions to allow visitors to find apartments based on location and features (much like apartments.com). I built a 'portal' function to allow ERS to manage a resident portal for each of their communities, including a calendar of events and maintenance requests. In 2011, because apartment shoppers are typically on the road while searching for an apartment, ERS pushed me to implement a mobile-optimized version of the website for mobile devices. Since then, I add mobile-optimized versions to all the websites I build.

In 2010, I built a website for Federal IT Security Institute (FITSI), an association founded by Jim Wiggins, my cousin Amee's husband. The website included ActiveK's Membership Management functionality, including registration, renewals, payments, and bulk email. I also built custom web applications to allow FITSI members to register and pay for exams and events, and to allow FITSI to manage members' progress toward FITSI certification. Later on, we added an online store to allow FITSI to sell training materials to members. For security purposes, FITSI always hosted the site on their own servers, and in 2020, we moved the site into the cloud via Microsoft's Azure platform. I have always appreciated FITSI as a client because, as an organization devoted to IT security, they regularly

test the website's security, attempting a wide variety of potential hacking techniques. To date, they have found no security issues, which comforts me.

In 2011, I built a new website for Community Housing Network (CHN), a substantial agency devoted to helping people find affordable housing. Besides being good work for a good organization, my work included creating a sprawling website, including a variety of features for outreach and donations. As with ERS, I built a mobile version of the CHN website because for many of their constituents, a cell phone is their only method of accessing the website. For CHN, I also built access security functionality to allow CHN to control which CHN staff had rights to update various sections of the website. While this addressed the functionality that FSU needed, I never built a full workflow management solution - because ActiveK was rarely deployed in such a multi-user environment.

In 2012, I built a new website and web application for the American Osteopathic College of Anesthesiologists (AOCA). The website included ActiveK's Membership Management functionality, including membership registration, renewals, payments and bulk email. I also built custom web applications to allow members to record Continuing Medical Education (CME) activities, and to allow 'resident' members to log their resident program activities and 'program directors' to evaluate residents' progress. Another web application allowed AOCA staff to manage AOCA meetings and conferences, to create schedules for events, manage registrations, and accept payments. In 2014, AOCA engaged an association management company who moved the website to another platform. It's a bummer when a client moves on, abandoning the great works I've done for them, but it's inevitable.

CandyPress Websites

After finding a worthy e-commerce solution in the CandyPress software, I built a couple dozen online store websites using CandyPress. As with ActiveK, the functionality already included in CandyPress allowed me to tackle ever more sophisticated website projects, and each project then demanded that I develop more functionality, which I built generically into the software, allowing me to tackle yet more sophisticated projects. I had to keep my updates delineated within the CandyPress code so that I could apply my updates to new versions of CandyPress as they came out.

In 2009, I built a CandyPress website for Fourgreen Auto Parts (FG) which sold Hyundai and Kia auto parts worldwide. The requirements for this online store were more challenging than for any other online store I have built. The greatest hurdle was that FG wanted to offer more than 1 million items via the store. To that point, the largest online store I had created held no more than a few hundred items. It took extensive work to modify CandyPress to accommodate so many items. In particular, I had to optimize all the product search and product management functions within the CandyPress software because their existing code was incredibly inefficient/slow (not an issue for a small store, but a big problem for such a large store). I also had to build a complex import function to load the CandyPress database from data files FG provided me once or twice a year.

Because FG sold worldwide, and shipped from warehouses in Korea and Detroit, I had to build functions to calculate shipping costs based on both where items would ship from and the shipping destination, and to accommodate 'deferred' shipping for items not currently in stock. Over time, I

integrated the website with Teapplix for managing shipments, and with Twilio for sending text messages, and with Xigla for customer support, and with a series of currency conversion rate suppliers (so customers could see prices in their native currencies), and with eBay to import orders made via eBay. I worked with 3rd-party developers to integrate the website with Square for credit card processing and Sellbrite for multi-channel inventory management.

In 2021, Fourgreen lost their rights to sell Hyundai/Kia-branded parts, so they closed down the website and set up a Shopify website to sell a few hundred aftermarket parts. For over 10 years, FG generated substantial work for me on an almost monthly basis, including everything from basic support and enhancements, to building major new functions, to implementing an ongoing barrage of Search Engine Optimization (SEO) techniques.

Florist Websites

In 2007, I was swimming with the local masters swim team (Farmington Hydrofoils), and one of my fellow swimmers (Mike Etienne) approached me with an opportunity. Mike was also in the software business, in his case developing software specific to the florist industry. A few of his customers (Strange's in Richmond, Virginia, Cathy Cowgill Flowers in Canton, Ohio and Neubauers Flowers in Uniontown, Pennsylvania) had websites provided by FTD and thereby limited to the capabilities of FTD's e-commerce solution, which did not address many of their needs. They wanted websites independent from FTD so they could add functionality beyond FTD's system and move forward faster than FTD's development schedule. They banded together to provide the specifications for the websites and to split the development costs. Later, a fourth florist joined the group, Adgate Flowers in Warren, Ohio.

Starting with a foundation of the CandyPress software, I added extensive functionality to address the unique requirements of the florist industry. Product search includes options to search by 'occasion', 'color' and 'price range'. Product listings include options for 'add on' items (e.g. balloons and candy). Store checkout includes 'delivery date' and a rules-based 'delivery charges' function to set rates based on where/how an order is to be delivered and whether it's to be delivered near a holiday (florists get very busy as we near Valentine's Day or Mothers' Day). I also added functions to allow customers to lookup addresses for local hospitals and funeral homes, and to 'clone' prior orders (when sending the same order to multiple recipients). Another quirky feature required by florists is 'guest checkout', whereby the checkout doesn't retain customer information in an online account - to accommodate discretion for those who might send flowers to someone other than their spouse.

The florists were also obsessed with Search Engine Optimization (SEO), so I added SEO functionality into the system. Before people ordered flowers online, they called a local florist and ordered flowers to be delivered anywhere. Still today, such orders are submitted into florist networks like FTD and Teleflora which assign them to a local florist near the delivery destination, and the florist who originated the order gets a share of the proceeds, as does FTD/Teleflora. As florists moved to selling online, any florist anywhere could accept orders to be delivered anywhere, submit the order to FTD/Teleflora, and get their share of the purchase - without really doing anything. As a result, Search Engine Optimization

(SEO) became essential for florists, as did Pay-Per-Click (PPC) advertising (e.g. Google AdWords). Some enterprising florists (some might refer to them as greedy) took this to extremes.

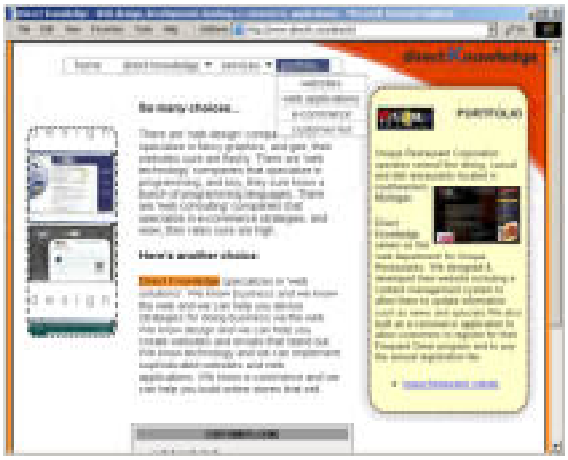
Wesley Berry (WB), a second-generation Detroit-area florist, built a huge operation (including a substantial call center in the back of a florist shop) to take in orders nationwide. He became expert in exploiting SEO and PPC so that his website and online ads would appear when people anywhere in the country searched for a florist. He developed a nationwide database of hospitals and funeral homes, and created web pages for each of them, so that if you searched for a hospital or funeral home, a WB page would be one of the first hits displayed. This annoyed other florists who were paid less for local orders than if customers called them directly, and it annoyed customers because WB often took orders that could not be delivered (perhaps a local florist could not produce the purchased arrangement or could not deliver when the customer requested), and it annoyed FTD and Teleflora who had to deal with all the annoyed florists and customers. Eventually, WB's aggressiveness resulted in so many customer complaints that he struggled to maintain relationships with credit card processors (too many disputed charges), and in 2016, WB shut down most of its stores and websites.

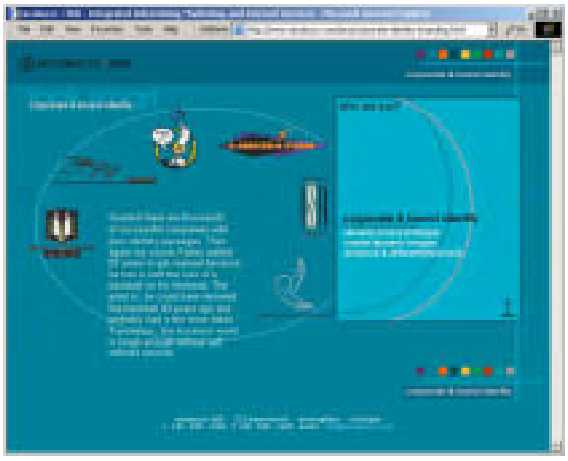
I describe all this because Wesley Berry twice approached me to assist with his website work, though he didn't hire me either time. After the company disappeared, I did end up working with the remnants of the Wesley Berry organization. Some WB debt holders wanted to exploit the assets they had extracted from the carcass of the old company by starting a new florist business: BloomsNationwide.com. I built an online store for them based on the software I built for the other florists. However, less than 6 months after the new website went 'live', they liquidated the company.

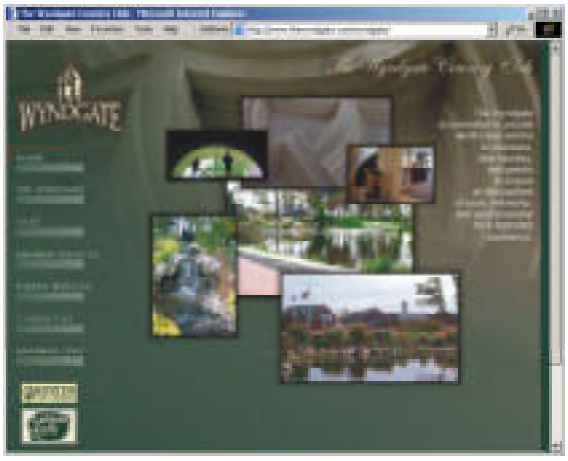
For a time, I imagined I might become a florist website specialist, managing websites for dozens of florists across the country. Probably because I did no marketing to florists, that did not happen. For varying reasons, Strange's left me in 2014, Cowgill's in 2015 and Neubauer's in 2017. As of 2023, I still manage the Adgate Florist website.

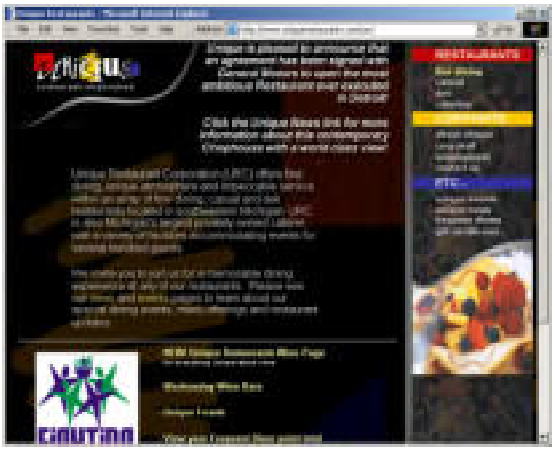
directKnowledge

Website Design & Development



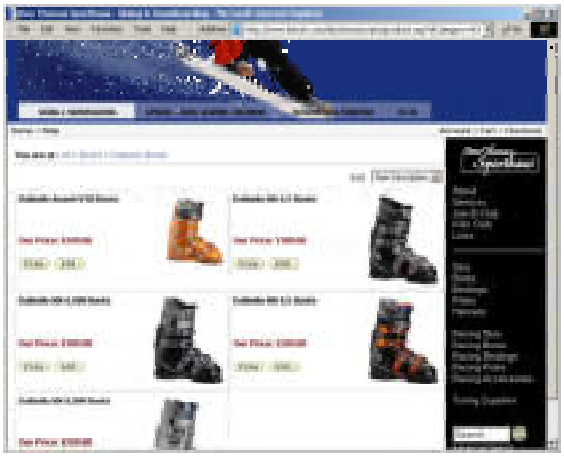


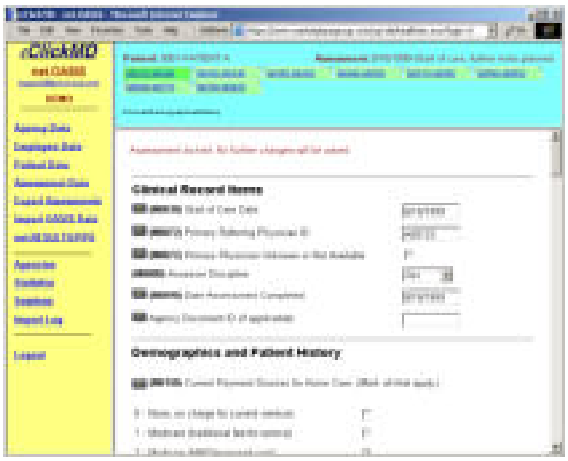


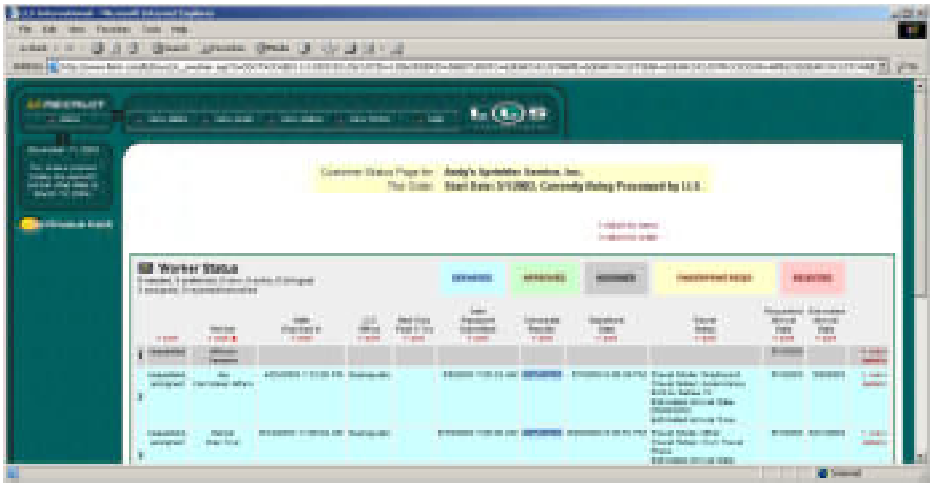


directKnowledge

E-Commerce Websites and Web Applications







directKnowledge

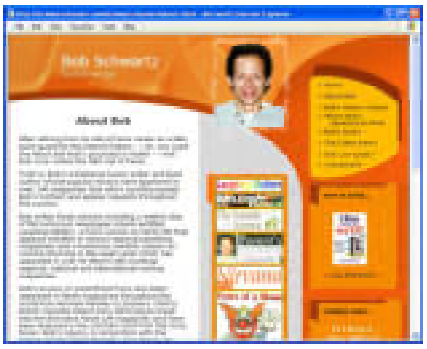
Recent Projects

XPLENS.com

We rebuilt this site based on a powerful new online store system with customer accounts, real-time shipping cost calculation (via UPS) and real-time credit card processing. The store includes a 'design your own lens' function to allow customers to create their own lens designs.

SchwartzHumor.com

We are in the process of building this new website. The project includes a mechanism to capture email addresses and to send out a monthly email newsletter.

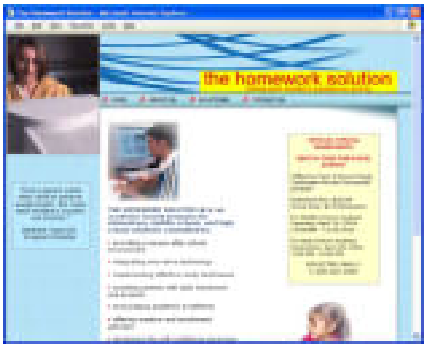


TroyStreetLofts.com

We built this new website and we are in the process of building a companion website for a related development project.

TheHomeworkSolution.com

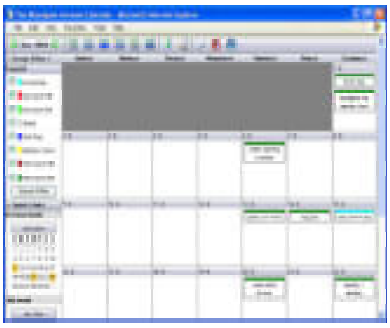
We built this new website.



Recent Projects

The Wyndgate Country Club

We developed an Intranet application to improve communications between various departments and people at The Wyndgate. The Intranet includes a powerful calendar application that allows Wyndgate staff to post events and to view events by one or more categories.



Rocky Peanut Company

We rebuilt the Rocky Peanut website to complement their new corporate identity & branding. The new site is based on a powerful new online store system with customer accounts, real-time shipping cost calculation (via UPS) and real-time credit card processing. Here are before and after shots:

GSA International

We developed an online order management system for this shipping company. The system enables a process flow whereby customers enter orders, GSA reviews and Prices the orders, the customer authorizes the priced order, and then GSA handles the shipment. All of these processes are managed via email. The system provides extensive order management and reporting functionality.

Recent Projects

Gravity Bar & Grill

We developed a new website for this fine-dining restaurant in downtown Milford. The website includes an email list to allow Gravity to send bulk email to their customers.

www.gravityrestaurant.com

StraightLine Advisors (SLA)

Direct Knowledge is collaborating with The Providers (Mike Morris) on a series of projects for SLA. First, we have moved their existing web system to a new, much less expensive server. Next, we're building a brand new system to replace the existing one.

www.myrplan.com

Summit Sports

When Don Thomas Sporthaus was bought by Summit Sports, we helped Summit rebuild and rebrand the website as Skis.com. This online store served thousands of customers during the holiday season. We also helped Summit make improvements to their SnowboardFusion.com website.

www.skis.com

Fastube

We are in the process of developing a new website for Fastube. The goal of the project is to create a more professional website that will put Fastube in a more equal footing with its competitors.

www.fastube.com

Recent Projects

Business Advisor Online (BAO)

We developed a full-featured 'portal' for this startup web company.. BAO provides hundreds of useful business articles within an easy-to-use framework to help business owners answer questions on a variety of subjects such as human resources, marketing and technology.

On the 'front-end', BAO customers are offered articles selected for them based on their industry and the size of their company. They receive emails containing links to new articles and 'subject reviews' which provide in-depth coverage of a particular issue such as 'Valuing and Selling Your Small business'.

On the 'back-end', BAO staff make use of a full-featured website management system to:

1) manage all of the content within the website,
2) manage user and subscription information including credit card renewals
3) create and send emails to selected user groups

Recent Projects

The Wyndgate Country Club

We developed a member portal to allow The Wyndgate to better communicate with its members. The portal includes an online member directory, newsletter, calendar of events, and more, all driven by a custom Content Management System (CMS) so that Wyndgate staff can easily manage all the content and functionality within the portal.

www.thewyndgate.com

Meadowbrook Country Club (MCC)

We developed a system to display member statements online via MCC's member portal. MCC staff can export statement data from their club management system, and post it directly to the website, so that MCC members can view both month-end and current-month statements.

www.meadowbrookcountryclub.com

StraightLine Advisors (SLA)

We developed a customer portal to allow SLA customers to view their portfolio data via a secure web-based application. Portfolio data is collected from customers' 401K/retirement accounts (via a 3rd party company—CashEdge) and posted to client records stored within Salesforce.com, the leading web-based Customer Relationship Management (CRM) system, used by SLA to manage all of their customers' information.

www.myrplan.com

Sommers Schwartz (S2)

We just completed a website redesign for S2 to provide a strong new look, better branding, and easier access to information The website is based on a custom Content Management System (CMS) to allow S2 staff to update most of the content within the website.

www.sommerspc.com

directKnowledge

Recent Projects

Urban Life Development (ULD)

We redesigned the ULD website to provide a much stronger look and easier access to information. ULD develops lofts and other properties within Detroit's Eastern Market area.

www.urbanlifedevelopment

Grand Vistas Travel (GVT)

We are in the process of developing a new website for this travel agency and integrating the design with VacationPort, a 3rd-party travel company. VacationPort supplies all of the vacation offerings, but they are displayed within GVT's design/framework.

www.grandvistastravel.com

ProQuest Insurance (PQ)

We are in the process of redesigning the PQ website to provide improved appearance, easier access to information, and to solve problems such as poor search-engine performance. ProQuest sells a variety of insurance products to law firms.

www.proquestinsurance.com

Reds on the River

We designed this 'placeholder' website to serve as an announcement for this soon-to-be-open restaurant in Rockford, MI (opening Aug. 2006). We'll be developing a 'permanent' website for Reds after the restaurant is open.

www.reds-live.com

directKnowledge

Recent Projects

Skiers Peak
Website Upgrade, Search Engine Optimization (SEO)

We are implementing an SEO project for Skiers Peak, with the goal of vastly improved search engine rankings for this online winter sports retailer. This entails upgrading the website software to display 'the right words' in 'the right places' throughout the website, and to allow Skiers Peak to better manage all of the website content. **www.skierspeak.com**

Here to Help Foundation
New Website

We designed a beautiful new website for this new charitable foundation.
www.heretohelpfoundation.org

Gala, A New American Bistro
New Website, Content Management System (CMS), Email Marketing

We designed a beautiful new website for this fine-dining restaurant in downtown Farmington. The website is based on a CMS that allows the restaurant staff to update most of the content within the site. We also implemented an email marketing system to allow Gala to send weekly emails to customers.
www.galafarmington.com

Pande Associates
Website Redesign, Email Marketing

We redesigned this website to be more professional, more 'global', and to provide substantially more content than the prior website. We also implemented an email marketing system to allow Pande to send newsletters and other emails to customers.
www.pandeassociates.com

directKnowledge

Recent Projects

Chemsol

Search Engine Optimization (SEO), AdWords

We developed an application to allow Chemsol to create descriptive pages for each of their many specialty chemical products. The primary objective was to improve their search engine rankings by including many of the words that their customers might type into a search engine 'search' box when seeking Chemsol's products. We also implemented Google AdWords campaigns for several of their products.
www.chemsol.com

StraightLine Advisors

Web-based Billing Application,
Salesforce.com integration

We developed a web-based billing system to replace a burdensome set of systems including QuickBooks and an outdated database application. The new system handles invoicing, payments, credit card payments, and more. The system is web-based, so that it can be accessed from anywhere, and it is integrated with Straightline's Salesforce.com application so that all of their customer data can be stored and viewed in one place.

The Burn Machine

E-Commerce Application Customization

We developed numerous customizations to TBM's online store system including:

- 'carryout' system to allow them to process 'cash & carry' sales at trade shows
- 'affiliate' system to allow them to provide discounts and track sales for 'affiliates'
- new reports to provide better sales tax reporting

www.theburnmachine.com

Content Management Systems

Direct Knowledge has developed a web portal designed to meet many of the communication needs of membership organizations such as associations, churches, neighborhood associations, and private golf/country clubs. The portal can be managed entirely by the staff of the organization via a secure web-based administration system. Organization members can login with individually assigned IDs and passwords to view a wealth of information about the organization. Key features include:

Content

- Article 'categories' provide a web section for each 'topic area' or 'group' within the organization.
- Articles can include formatted text, images, links.
- Article start and end dates allow content to appear and disappear based on the calendar.
- Integrated full-text search
- Articles can be tagged as 'public' or 'private' to create two websites: one for the public and one for registered users.

Member Functions

- Member directory/roster with search and sort functions allows easy access to member info.
- Members can update their own profiles.
- Bulk email system for all users or selected groups of users.
- Survey/vote system collects member opinions.

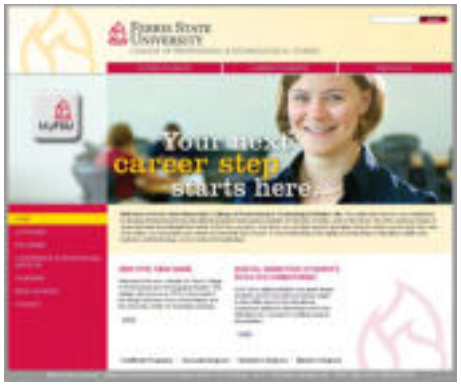
Calendar

- Events are treated as 'articles', including categories and public/private events.
- Event descriptions can include formatted text, images, links.
- Color coded calendar based on categories.

Recent CMS Projects

Ferris State University

College of Professional and Technological Studies



We are developing a new website for FSU-CPTS which will be going 'live' within the next couple of weeks. This is an entirely 'public' site, to be managed by FSU-CPTS marketing/communications staff. The CMS allows non-technical staff to manage a rather large, quickly evolving website, using an easy-to-use system rather than HTML coding.

fsu.directk.com (temporary site)

Reynolds Water Conditioning

We are developing a new website for Reynolds which will be going 'live' within the next few weeks. The website includes a 'public' website managed via the CMS and also an online store.

reynoldswater.com (existing site)
reynoldswater.com/portal.asp (new site)

The Wyndgate Country Club

This is a 'private' site used by The Wyndgate to communicate with club members and to allow club members to stay better connected with club happenings and other members.

directKnowledge

Content Management Systems

Web Applications

Offersubmission.com

We developed a web-based system to help banks sell off their portfolios of foreclosed homes. Unlike the 'old way', this website directly connects buyers and sellers so that sellers see all of the offer activity for each home. This dramatically speeds up the process and generates higher average sales prices.

www.offersubmission.com

J&B Medical

We are developing a customer service application for J&B's diabetic customers. This website allows the customers to update their profile information, pay account balances, and confirm their monthly diabetic supplies orders.

This application interfaces with J&B's internal customer management system.

To facilitate management of a broad array of customer activities, an administrative system presents a dashboard for each J&B user allowing them to view and manage a selected set of activities.

directKnowledge

Custom Web Applications

Weaver Media

WMOnline - Order Management System

We developed a web-based order management system, to allow Weaver's customers to place orders for advertising & signage, to allow Weaver staff to manage the fulfillment of the orders, to allow Weaver's suppliers to receive orders and update order status, and to allow advertisers to view advertising/signage by facility and date range.

Federal IT Security Institute

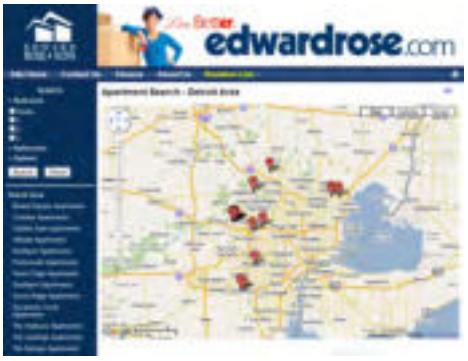
Membership / Portal System

We developed a web-based portal to support a variety of membership activities for FITSI, including membership management, exam/event registration, certification management, and member communication.

Edward Rose & Sons

Property Management System

We developed a web-based Property Management System (PMS) to drive the EdwardRose.com website. The PMS allows Edward Rose staff in 4 offices to manage all of the information for each of their apartment communities, including property information, slideshows, photo galleries, PDF/printable brochures, interactive Google-based maps, and online rental applications.



In addition, we implemented portal websites for each of the more than 100 Edward Rose apartment communities, based on our ActiveK Content Management System (CMS). These sites allow Edward Rose to deliver information to community residents and allow residents to submit service requests to Edward Rose.

directKnowledge

Custom Web Applications

OfferSubmission.com

We developed a web-based system to help banks sell off their portfolios of foreclosed homes by allowing bank staff to view/manage all of the offer activity for each home. Recently we have added workflow management features to make it easier for the banks to manage the large number of properties and offers within the system.

E-Commerce Websites

Fourgreen.com

This website offers over 750,000 auto parts for sale around the world. Custom features include import routines to load products and pricing from vendor-supplied databases, inventory management functions, custom shipping cost functions, and integration with 3rd party shipping management solution. Recent updates include Search Engine Optimization (SEO) and addition of Aftermarket products to the store.

AdgateFlorist.com, CathyCowgillFlowers.com, NeubauersFlowers.com, Stranges.com

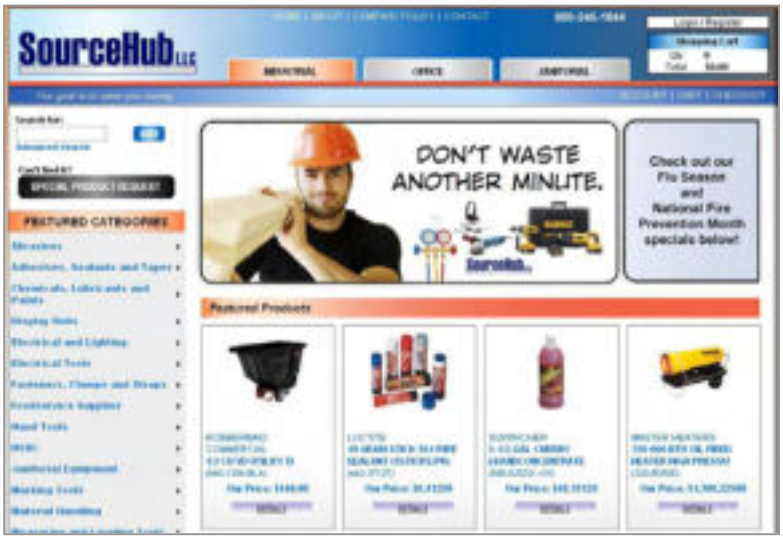
These websites generate hundreds of thousands of dollars in annual sales of flowers and gifts. Custom features include delivery options, order add-ons such as balloons and bears, and the ability to quickly change featured products to address seasonal holidays like Valentine's Day.

Sourcehub.com

This website offers over 100,000 items in three major categories: Industrial, Office and Janitorial, to serve as a one-stop shop for many businesses. Custom features include import routines to load products and pricing from a variety of vendor-supplied databases.

What we do:

- websites
- ecommerce sites
- web applications
- content management systems (CMS)
- intranets
- search engine optimization (SEO) and Pay-Per-Click (PPC) advertising



Direct Knowledge, L.L.C.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

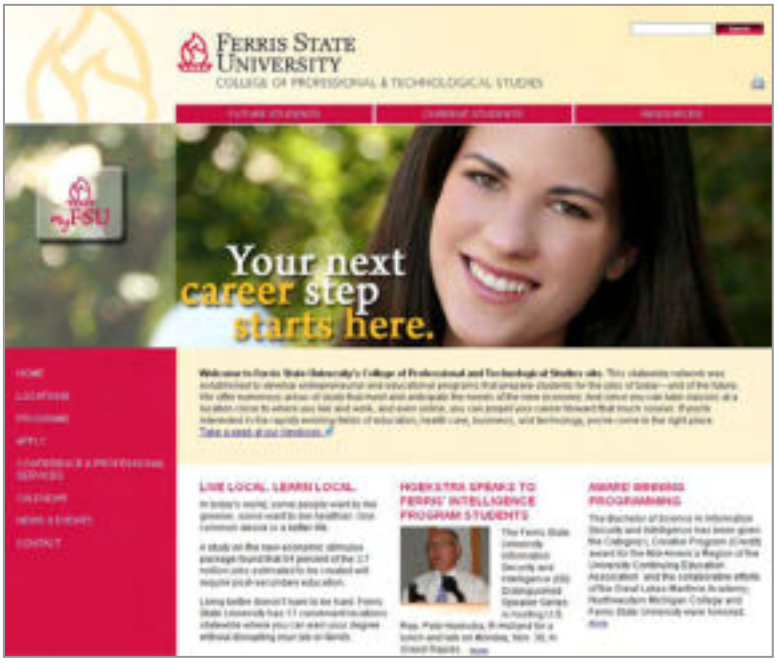
EMAIL: info@directK.com
WEB: www.directK.com

Why use DirectK?

We do the right things. I work with you to understand what you're trying to accomplish and then design the project to help you achieve those objectives. This seems obvious, but I have seen many projects that weren't done this way, and they spent a lot of time and money building 'the wrong thing'.

We do things right. A website needs to look good, work correctly, and make it easy for people to do what they need to do, whether it is 'get information' or 'buy something' or 'use a web application'. In addition the website needs to be fast, reliable, secure, easily updated, easily enhanced, and more. This also seems obvious, but there are so many websites out there that simply don't work 'right'.

Experience. I do most of the work and I've been building websites for 15 years and software for over 25 years. I've learned many lessons along the way, about how to build the right things, how to build them right, and how to build them inexpensively.



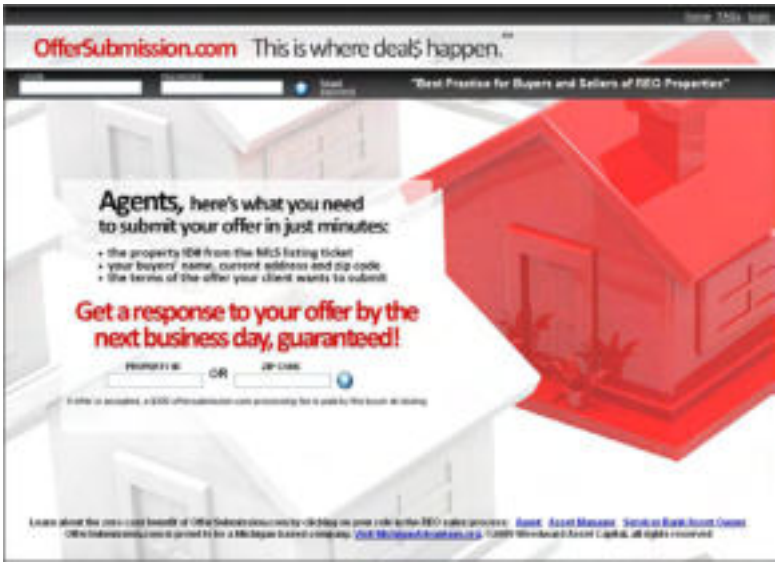
Direct Knowledge, L.L.C.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Inexpensive. Relative to the competition, DirectK is a bargain. I have been in numerous bidding situations where my price was dramatically lower than the other bids, sometimes as low as 50% of the other bids. And given that my proposals are based on the customers' real requirements, there much fewer 'add-ons' than with many other vendors.

On Budget and On Time. All of our projects are fixed-price based on agreed-upon specifications and all of our proposals include a project timeframe. Our goal is no surprises on the bill or the completion date.



Direct Knowledge, L.L.C.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Good referrals:

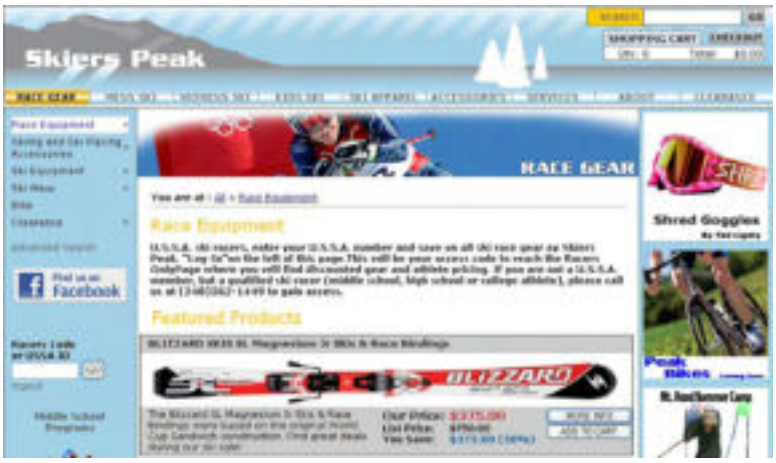
- Companies/organizations looking to update the design and/or functionality of their websites.

- Companies/organizations who want to sell things online.

- Companies/organizations who want to put a customer or business process on the web, like customer service applications.

- Companies/organizations looking to improve the number and quality of visits to their websites (SEO/PPC)

- Companies/organizations with very dynamic website requirements, who need a content management system to drive the website. In particular, membership organizations who need to communicate both to the public and to their members.



Direct Knowledge, L.L.C.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

CMS 101

Content Management Systems (CMS) allow a website owner to update content within their website via easy-to-use tools. Here is the Network Partners home page, along with the CMS page used to update the page.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

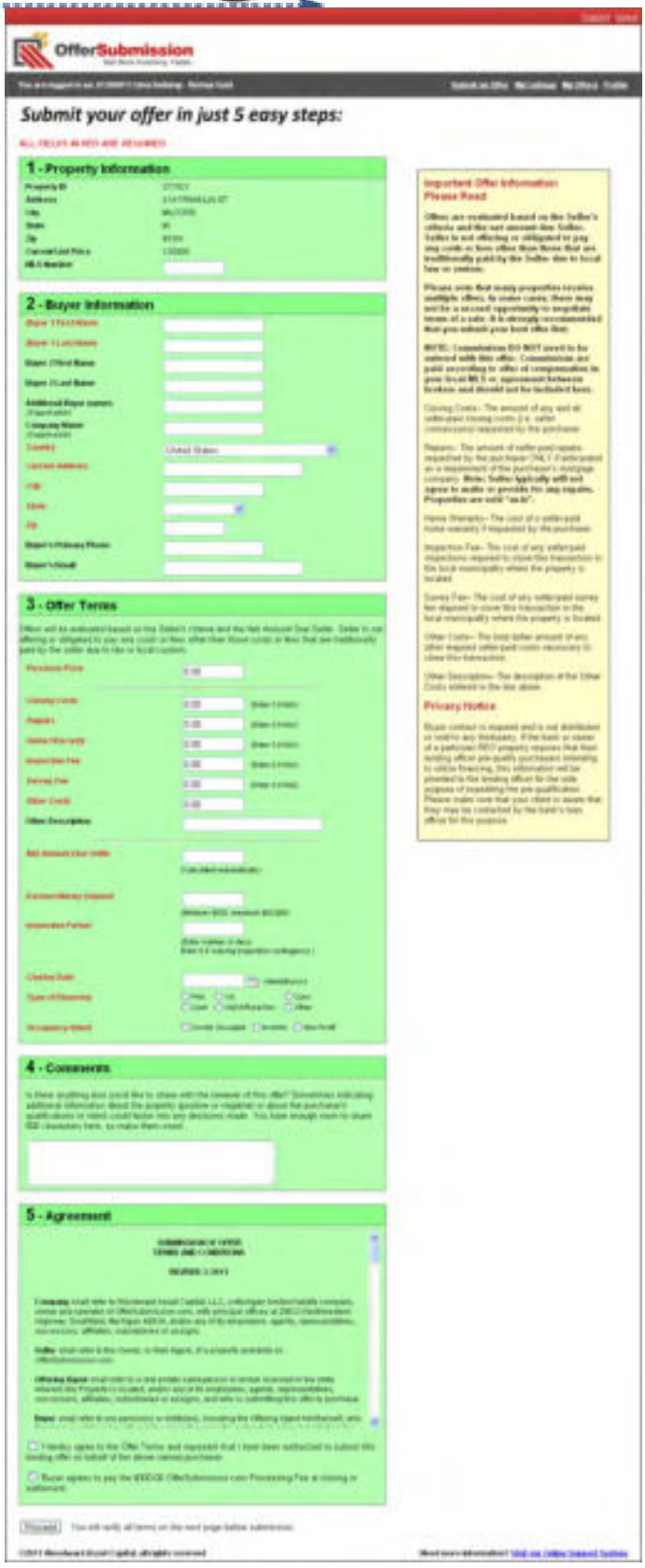
EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

Web Applications 101

Web Applications are software systems that run via the Internet and using a web browser as the mechanism for interacting with the software. Technically, a CMS is a 'web application'.

Often, we integrate a full-featured CMS into the web applications that we build, so that all of the text in the web application can be edited by the website owner – for example, the text in the yellow box at the right of the form.



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

E-Commerce Systems

E-Commerce systems are ‘web applications’ that have the additional feature of accepting credit card transactions via the Internet, which requires extra security measures and also integration with other systems for credit card processing.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

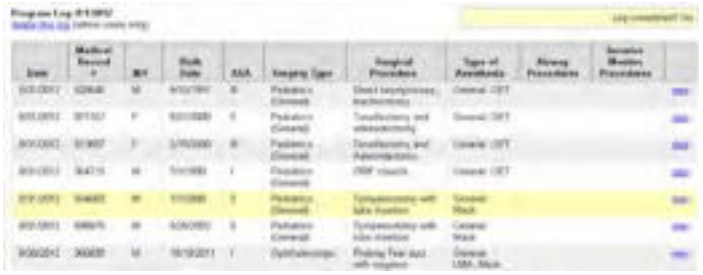
CMS 301

Content Management Systems (CMS) allow a website owner to update content within their website via easy-to-use tools.

DirectK specializes not only in CMS websites, but in 'advanced' CMS websites.

For **AOCA**, we implemented a basic 'public' website, plus a 'members only' website (accessible after logging in), plus numerous custom functions including:

- Member dues system with reminder emails and online dues payment.
- Membership application system with document upload (for diplomas, licenses, etc.) and a process for AOCA to review and approve applications.
- Event calendar and registration system to allow attendees to register and pay for various sessions.
- Residency 'program logs' system to allow anesthesiology residents to manage detailed logs of all of the procedures they perform each month and for the year.
- Evaluation system to allow residents to evaluate the residency program and program directors to evaluate residents.



For other clients we have implemented custom functions including:

- **Edward Rose & Sons**
  A property management system to allow ERS staff to manage all of the information about their extensive portfolio of apartment communities.
- **StraightLine**
  A portfolio management system to allow both staff and customers to view financial portfolio information and investment recommendations.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

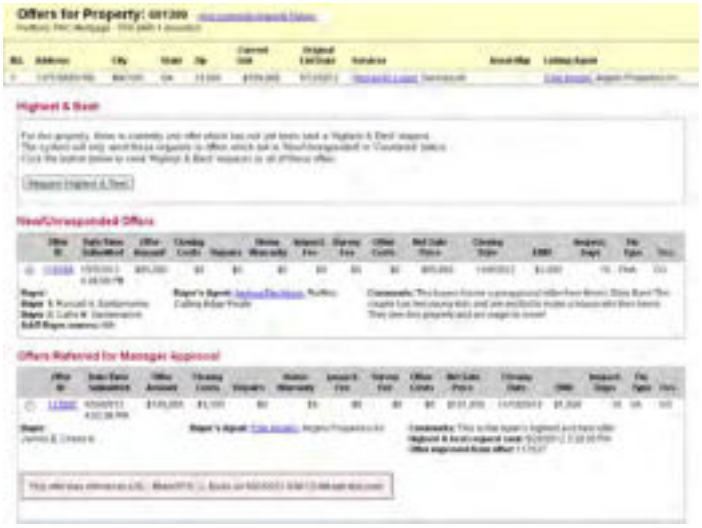
Web Applications 301

Web Applications are software systems that run via the Internet and using a web browser as the mechanism for interacting with the software. Technically, a CMS is a 'web application'.

DirectK specializes not only in Web Applications but in 'advanced' Web Applications.

For **Offersubmission**,for the past 4 years, we have worked on an ever-evolving system to help banks sell homes from their portfolios of foreclosed homes: For the past year, we have been working on a similar system to help banks/homeowners sell homes via short sale. These systems include features such as:

- Many layers of property/offer status management and email notifications – to make sure that no-one drops any balls and offers proceed quickly to closing.
- Approval rules system to specify various classes of users and the rights that they have to handle various offers based on the quality of each offer (and other parameters) including referring offers to higher-level users.
- Offer history system to track every activity for an offer including who, what, when and notes to indicate why.
- Reporting mechanisms to present performance statistics for various properties and users, such as monitoring average response time for bank staff handling incoming offers.
- Document management systems to allow the parties to an offer to upload, approve, display and print property and offer documents.



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

E-Commerce Systems 301

E-Commerce systems are ‘web applications’ that have the additional feature of accepting credit card transactions via the Internet, which requires extra security measures and also integration with other systems for credit card processing.

DirectK specializes not only in E-Commerce Systems but in ‘advanced’ E-Commerce Systems.

For **FourGreen**, we implemented an online store offering over 500,000 auto parts for sale worldwide. In addition to the typical online store functionality, we implemented numerous custom functions including:

- Shipping method/cost system based on shipments from warehouses in Detroit and Korea and destinations worldwide.
- Integration with a third-party shipping management system to help make sure that all orders are processed correctly and in a timely manner.
- Special orders system to allow for orders of non-standard parts and high quantity orders with special pricing.
- Shippable orders system to optimize the applications of available inventory to open orders, so as to complete orders as quickly as possible so that they can be shipped.
- Search engine optimization techniques so as to optimize all 500,000+ part pages for likely keyword searches.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

What we do:

- website design
- ecommerce websites
- web applications
- content management systems (CMS)
- search engine optimization (SEO)

Why use DirectK?

We do the right things. I work with you to understand what you're trying to accomplish and then design the project to help you achieve those objectives. This seems obvious, but I have seen many projects that weren't done this way, and they spent a lot of time and money building 'the wrong thing'.

We do things right. A website needs to look good, work correctly, and make it easy for people to do what they need to do, whether it is 'get information' or 'buy something' or 'use a web application'. In addition the website needs to be fast, reliable, secure, easily updated, easily enhanced, and more. This also seems obvious, but there are so many websites out there that simply don't work 'right'.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

Experience. I do most of the work and I've been building websites for over 17 years and software for over 25 years. I've learned many lessons along the way, about how to build the right things, how to build them right, and how to build them inexpensively.

Inexpensive. Relative to the competition, DirectK is a bargain. I have been in numerous bidding situations where my price was dramatically lower than the other bids, sometimes as low as 50% of the other bids. And given that my proposals are based on the customers' real requirements, there much fewer 'add-ons' than with many other vendors.

On Budget and On Time. All of our projects are fixed-price based on agreed-upon specifications and all of our proposals include a project timeframe. Our goal is no surprises on the bill or the completion date.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

Good referrals:

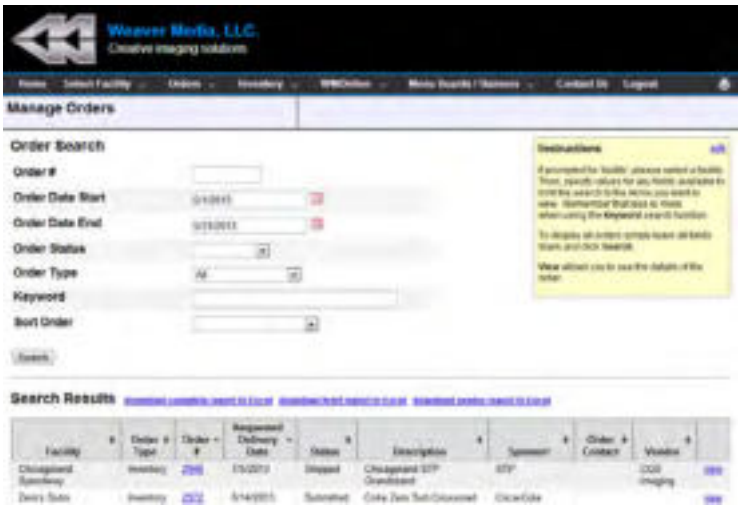
- Companies/organizations looking to update the design and functionality of their websites.

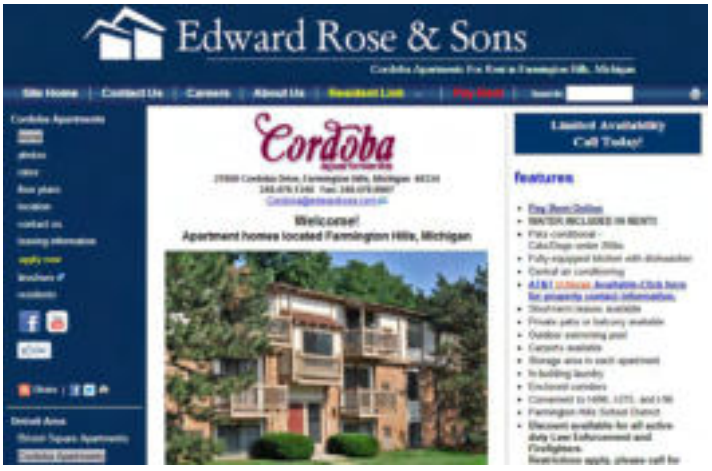
- Companies/organizations who want to sell things online.

- Companies/organizations who want to put a business process on the web, like customer service applications.

- Companies/organizations looking to improve the number and quality of visits to their websites (SEO).

- Companies/organizations with very dynamic website requirements, who need a content management system to drive the website. In particular, membership organizations who need to communicate both to the public and to their members.





Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

ActiveK

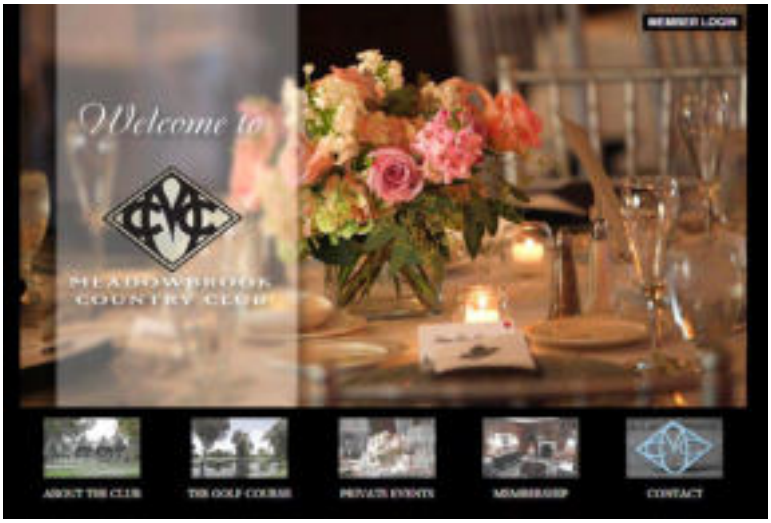
Many of the examples shown were built using Direct Knowledge's ActiveK Content Management System (CMS).

While the sites have a wide variety of designs and functionality, they all share the same core software.

The software is easy to use so that website owners can easily update the content on their websites, whether changing a few words on a page, or developing complete new sections/pages of the website.

It's search engine friendly, so without even trying hard, the website will be presented pretty well with the search engines, and it provides functions to allow you to get even better results.

The ActiveK software is under continuous improvement, to add more functionality, improve existing functions, and address new security concerns. Any of these ActiveK-powered websites can be easily upgraded to new version of the software, so as web technology progresses, so too can your website.



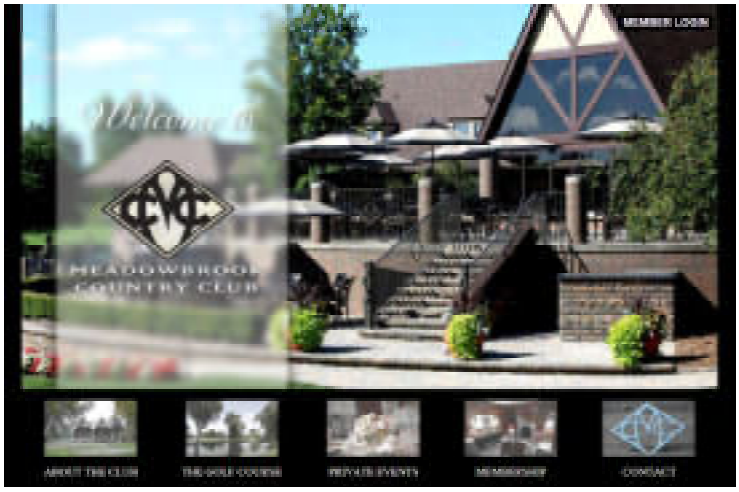
Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

What we do:

- website design
- ecommerce websites
- web applications
- content management systems (CMS)
- search engine optimization (SEO)



Recent Projects:

Meadowbrook Country Club
www.meadowbrookcountryclub.com

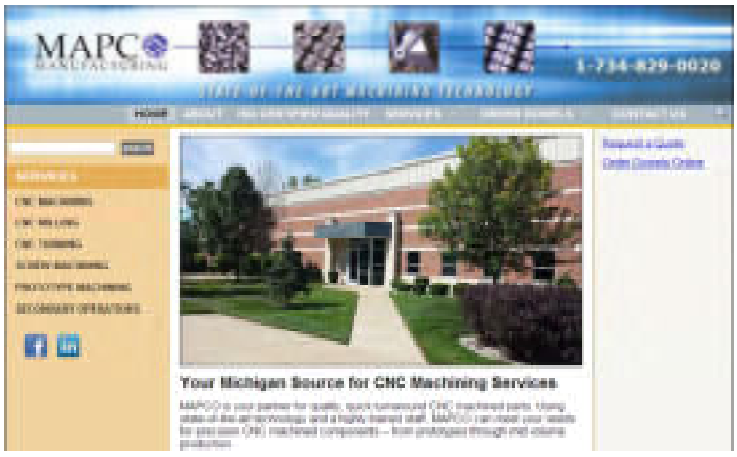
Redesigned website and did SEO work to attract more events (weddings, parties, meetings) to the club. Also added a mobile version of the website.

The Wyndgate Country Club
www.thewyndgate.com
wyndgate.directk.com

Redesigned website and did SEO work to attract more events (weddings, parties, meetings) to the club. New site not yet live, replacing a design circa 2000.

Mapco Manufacturing
www.mapcomfg.com
www.mapcomfg.com/activek/index.asp

Redesigned website and did SEO work. Added custom 'dowel selector' function to online store for easier product selection. New site not yet live, replacing a design circa 2007.



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

The WebAround
www.thewebaround.com

Designed and built a new ecommerce website with custom logo-upload and shipping fee calculation functions.

Here to Help Foundation
www.heretohelpfoundation.org
h2h.directk.com

I'm currently rebuilding this website with a new design, new content and new features including ecommerce functionality for donations. New site not yet live, replacing a design circa 2007.

Community Housing Network
www.communityhousingnetwork.org

I'm currently working on several website updates including custom ecommerce functions for donations and events and also the addition of several blogs.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Good referrals:

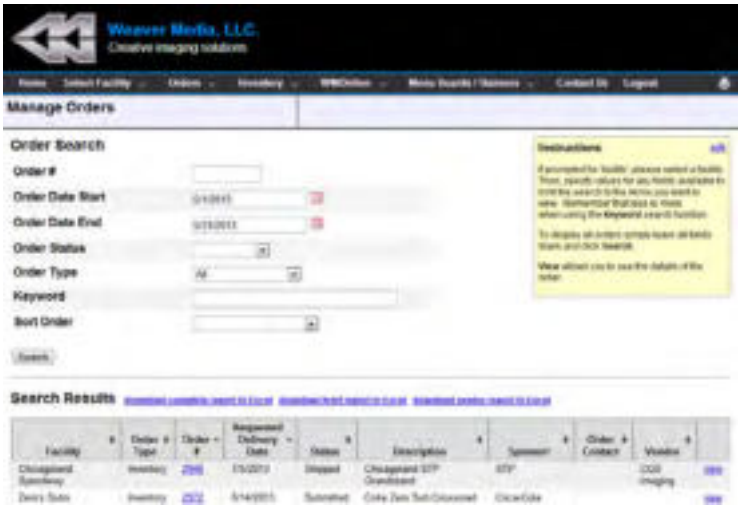
- Companies/organizations looking to update the design and functionality of their websites.

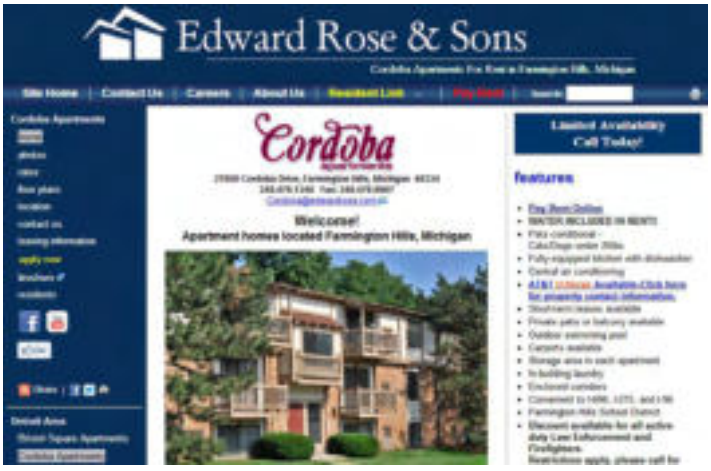
- Companies/organizations who want to sell things online.

- Companies/organizations who want to put a business process on the web, like customer service applications.

- Companies/organizations looking to improve the number and quality of visits to their websites (SEO).

- Companies/organizations with very dynamic website requirements, who need a content management system to drive the website. In particular, membership organizations who need to communicate both to the public and to their members.





Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

ActiveK

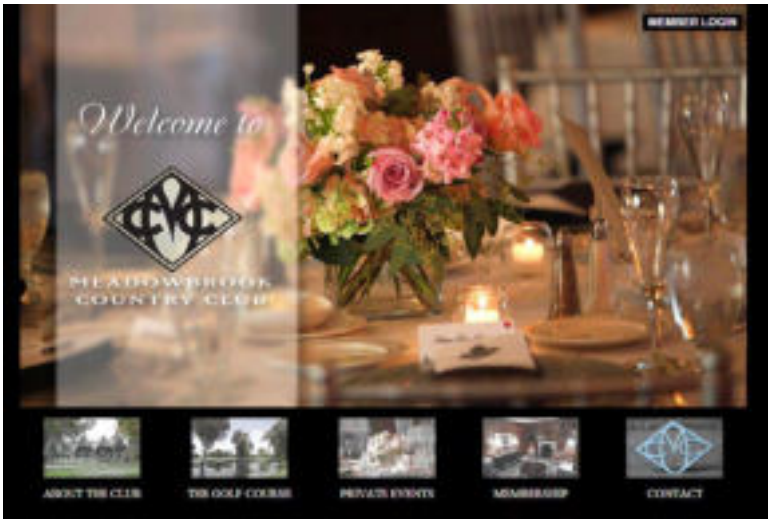
Many of the examples shown were built using Direct Knowledge's ActiveK Content Management System (CMS).

While the sites have a wide variety of designs and functionality, they all share the same core software.

The software is easy to use so that website owners can easily update the content on their websites, whether changing a few words on a page, or developing complete new sections/pages of the website.

It's search engine friendly, so without even trying hard, the website will be presented pretty well with the search engines, and it provides functions to allow you to get even better results.

The ActiveK software is under continuous improvement, to add more functionality, improve existing functions, and address new security concerns. Any of these ActiveK-powered websites can be easily upgraded to new version of the software, so as web technology progresses, so too can your website.



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

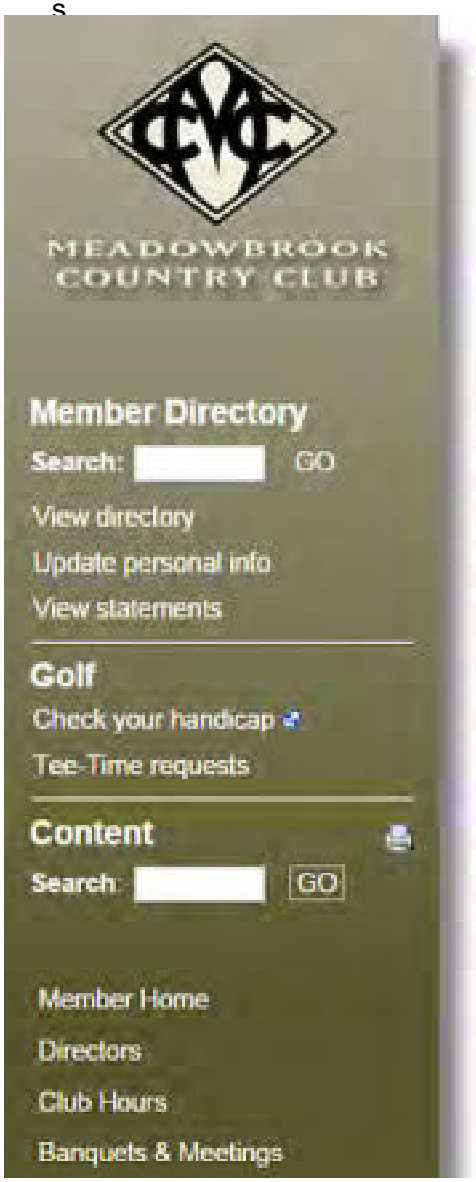
directKnowledge

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge



Statement

Statement Date
8/31/2014

Member Number
000022

Amount Due
$1,470.93

> select another statement

Date	Ref.	Description	Charge Amount	Gratuity	Sales Tax	Total Charges	Payment	Current Balance
		Balance Forward:						1,352.54
08/10	247900	FOOD/BEVERAGE	196.55	37.34	11.79	245.68		1,598.22
08/10	254300	FOOD/BEVERAGE	10.00	1.90	0.60	12.50		1,610.72
08/10	51037	First Tee	15.00	0.00	0.00	15.00		1,625.72
08/10	560900	POOL STAND	11.80	2.24	0.71	14.75		1,640.47
08/20	077000	FOOD/BEVERAGE	65.25	12.40	3.91	81.56		1,722.03
08/20	093000	FOOD/BEVERAGE	14.75	2.80	0.89	18.44		1,740.47
08/20	301800	13TH TEE	7.15	1.36	0.43	8.94		1,749.41
08/20	51421	First Tee	260.00	0.00	0.00	260.00		2,009.41
08/20		Payment on Account	-1,352.54	0.00	0.00	-1,352.54		656.87
08/27	577900	POOL STAND	33.80	6.42	2.03	42.25		699.12
08/29	51722	First Tee	30.00	0.00	0.00	30.00		729.12
08/29	620500	SWIMMING	12.00	0.00	0.00	12.00		741.12
08/31	581600	POOL STAND	15.85	3.01	0.95	19.81		760.93
08/31	F64804	DUES	540.00	0.00	0.00	540.00		1,300.93
08/31	F64805	CAPITAL DUES	150.00	0.00	0.00	150.00		1,450.93
08/31	F64806	DUES GOLF CRSE	20.00	0.00	0.00	20.00		1,470.93

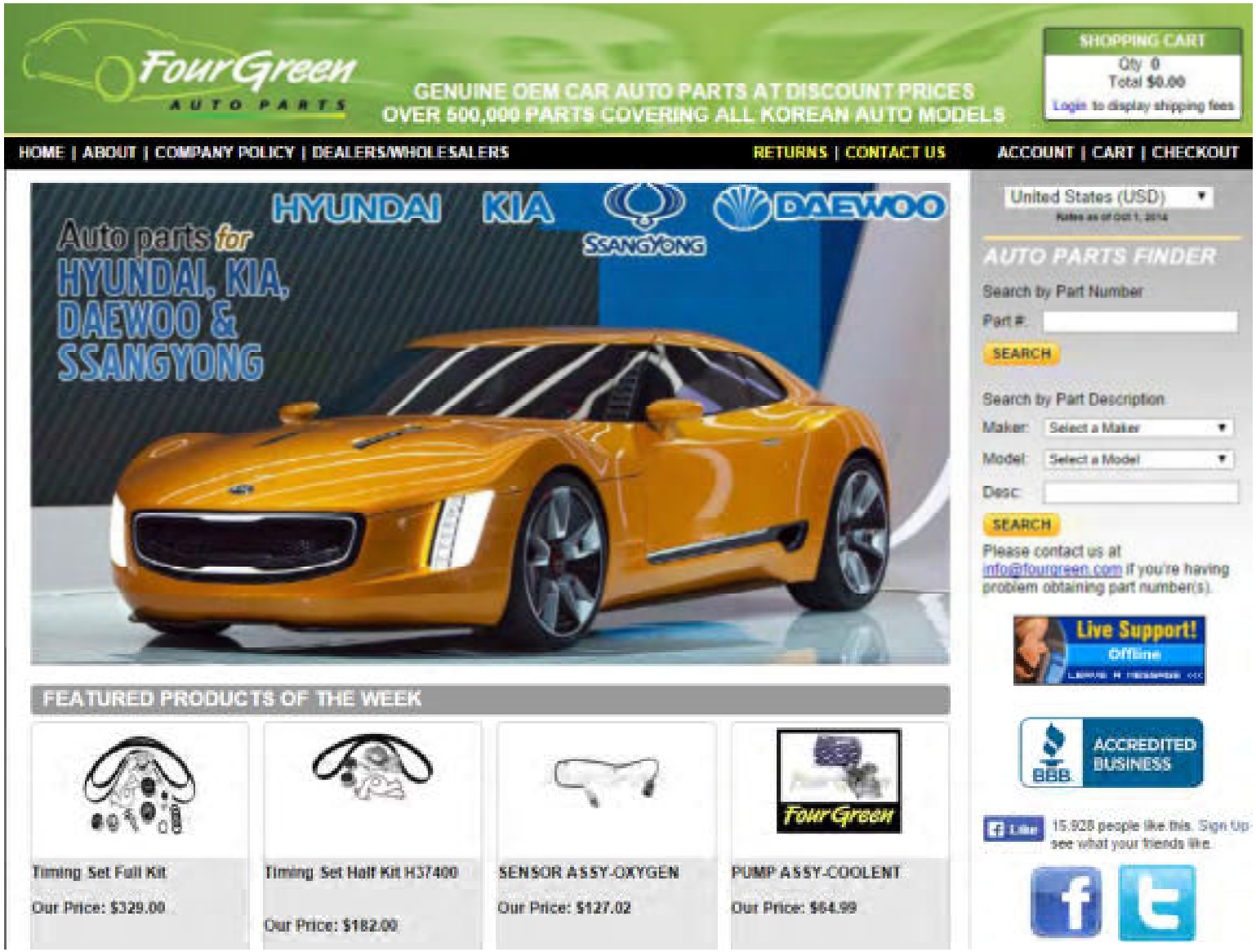
Unused Minimum	Current	30 Days	60 Days	90 Days		
0.00	1,470.93	0.00	0.00	0.00	Amount Due -->	1,470.93

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

Fourgreen - Manage Shippable Orders

preview application of inventory to orders

apply inventory to orders

Show Orders where Shippable Status is: Show all Orders — Find

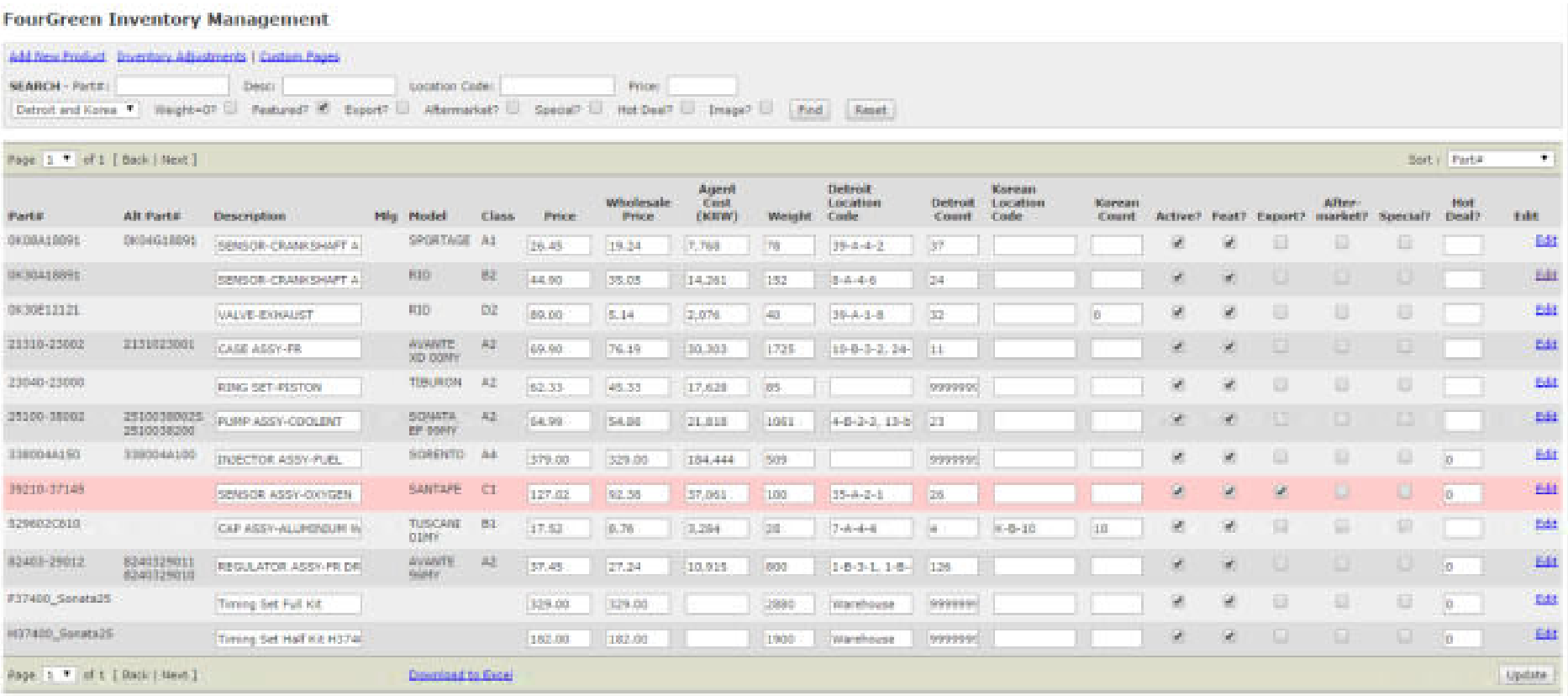
Show orders between: — and: — Find

Sort : Order (Descending)

Order Type	Order#	Date/Time	Name/Address	Total	Order Status	Shippable?	Part#	Description	Quantity	Assigned	To Be Applied
FG	126657	10/1/2014 8:46:21 AM	Rodriguez, Mauricio 11742 SW 12 street Miami, FL US	50.16	Paid	Korea 0 of 1	986201Y100	RESERVOIR ASSY-W/SHLD WASHER	1	0	0
FG	126591	9/30/2014 4:20:15 PM	Casseres, Eldon 7701 NW 15th Street, Suite # CW229394 Miami, FL US	66.16	Paid	Korea 0 of 3	838301F001 839101F000 839201F000	MOULDING ASSY-RR DR DELTA LH COVER ASSY-RR DR DELTA INR LH COVER ASSY-RR DR DELTA INR RH	1 1 1	0 0 0	0 0 0
FG	126583	9/30/2014 3:27:16 PM	De Vries, Deck 3048 sw 141st ave miramar, FL US	204.62	Paid	Korea 0 of 18	222244A000 223114A000 813111F020	SEAL-VALVE STEM GASKET-CYLINDER HEAD LATCH ASSY-FR DR LH	16 1 1	0 0 0	0 0 0
FG	126202	9/28/2014 3:27:24 PM	SANTILLAN TORRES, DIEGO J. E13B Psj A, N58-138 y Leonardo Murialdo QUITO, EC	148.20	Paid	Korea 7 of 29	1540710201 2522142400 3745239100 3746142000 8112637000A 85512H1001JI 8641036010 86440H1000 87702H1000 88710H1010JI 89376H1000JI 89543H1000LK 9262029610OJ 97613H1050	BOLT-EYE JOINT PULLEY-COOLANT PUMP BOLT BRACE-GENERATOR CLIP-HOOD INSULATING PAD MTG LID-TOOL BUMPER-HOOD OVERSLAM BUMPER-HOOD OVERSLAM CLIP-SIDE GARNISH MTG HANDLE-TILTING FOLDING MECHANISM ASSY KNOB RECLINER LAMP ASSY-LUGGAGE COMPARTMENT PULLEY-CRANK	1 1 1 1 10 1 2 2 5 1 1 1 1 1	0 0 0 0 7 0 0 0 0 0 0 0 0 0	0 0 0 0 7 0 0 0 0 0 0 0 0 0
FG	126120	9/28/2014 4:14:41 AM	Boldrev, Yury 95, 5, Kotlyarova str Krasnodar, RU	115.64	Paid	Korea 0 of 1	433704D600	HUB & SLEEVE-SYNCHRONIZER(S&R)	1	0	0
FG	126097	9/27/2014 1:54:39 PM	Mendoza, Angel 2462 Lennox Street Pomona, CA US	91.56	Paid	Korea 0 of 11	811252M000 8112637010	PAD-HOOD INSULATING CLIP-HOOD INSULATING PAD MTG	1 10	0 0	0 0
FG	126088	9/27/2014 9:40:37 AM	Doej, Harry Iggiaanut 3a Nuuk, GL	255.55	Paid	Korea 0 of 1	823062E0319M	PANEL COMPL-FR DR TRIM RH	1	0	0

directKnowledge

FourGreen Inventory Management



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

StraightLine

Personal. Unbiased. Proactive.

Client Login

StraightLine was created over ten years ago to provide plan design assistance, participant education, advice and management to universities, colleges, corporations and individuals. Our mission is to ensure that each and every employee and individual have the tools necessary to reach their retirement goals.

Higher Education
Integrated Plan Services

Expert Fiduciary care for your plan and participants.

401(k) / 403(b)
Managed Accounts

How do SMART people manage their 401(k) / 403(b) accounts?

They don't.

STARS
Investment Services

You've never known an investment firm like us - - we always act in the best interest of our clients.

Wealth
Management

Put your assets where you put your trust.

© 2012 StraightLine. All rights reserved.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

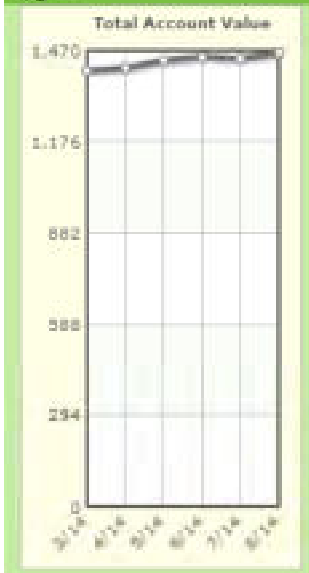
PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

StraightLine
Personal. Unbiased. Proactive.

logout | Home | My Portfolio | My Communications | My Profile | Market Commentary | Contact Client Services



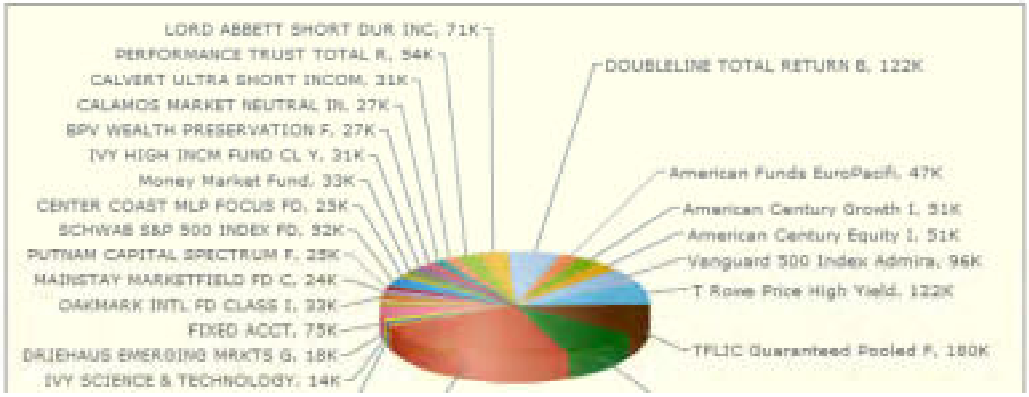
Portfolio Information

The information on these pages is gathered from your account custodian and is believed to be accurate, but cannot be guaranteed by StraightLine. We provide it as a service to you so you can see all of your account balances in one place.

In general the account data is from the previous business day's closing figures. On occasion the custodian or the program that gathers their information does not provide the data we need at the appointed hour and the results may be delayed for a day. There are a few custodians whose information cannot be updated daily and so values are only corrected at the end of each month.

If you find that your account data is not updating as you think it should, or if you have any other questions please give us a call at 866-401-5238.

Account Name	Account #	Type	Value
Fidelity NetBenefits - rollover	[illegible]	Pro (K)	$0.00
IRA	[illegible]	STARs - Schwab	$167,729.15
Lincoln Financial Group Annuities	[illegible]	Pro (K)	$75,268.33
TOD	[illegible]	STARs - Schwab	$467,468.58
Transamerica Retirement Solutions My TRSRetire	[illegible]	Pro (K)	$1,206,937.77
Total			**$1,917,403.83**



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

Billing - University Billing

« return to billing menu

Payments due as of: 9/30/2014

Generate payments in Salesforce

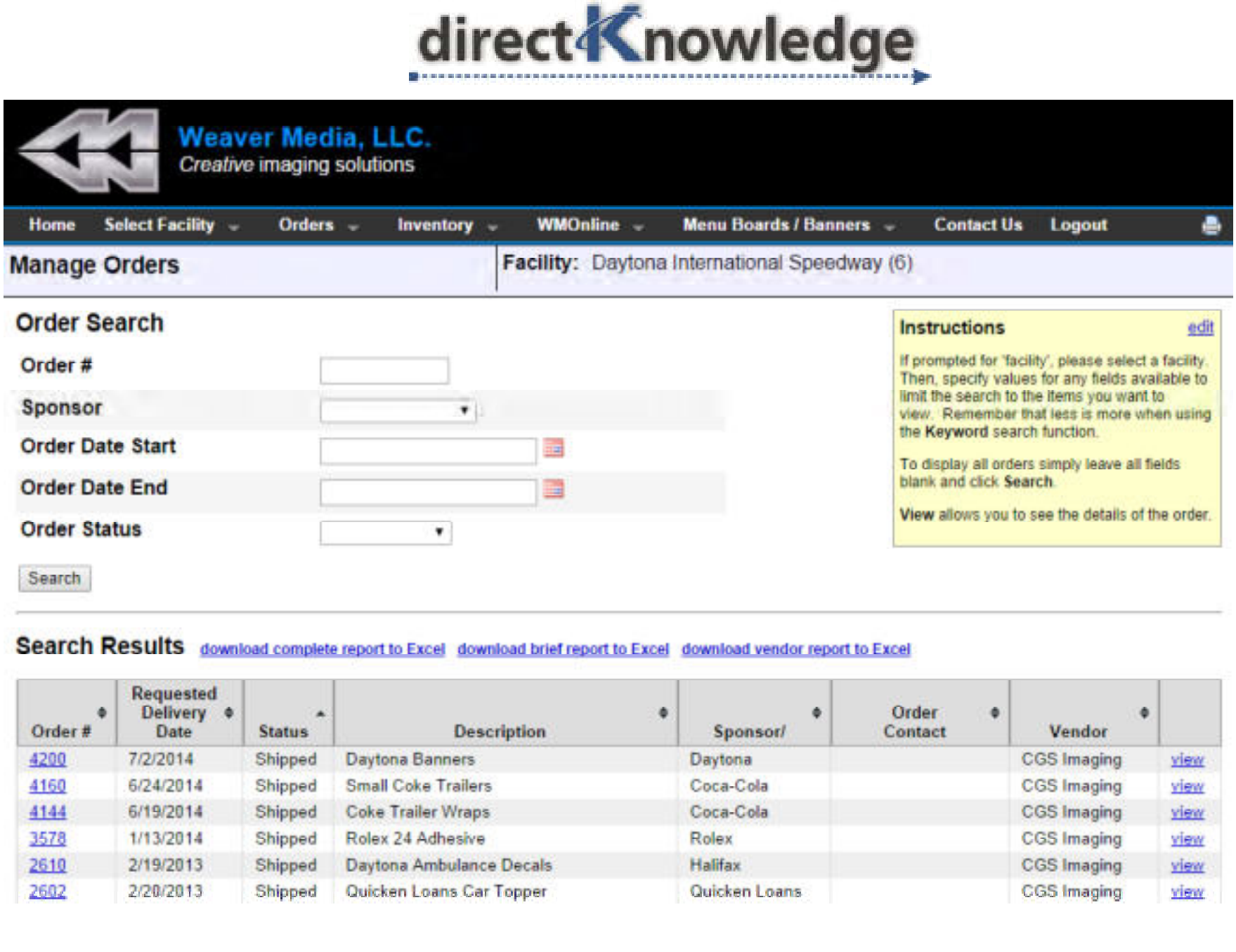
Export payment information to Excel

Plan	Client	PP Accept Date	Client Type	Months Invcd	Qtrly Amount Due	Pay Amount (prorated)	Most Recent Payment	Fid. elity	TIAA CREF	Van. guard	MSU 457	Comments	Account Details
NU	[redacted]	10/22/2013	Pro(k)	3	$125.00	$125.00	Payment made 7/18/2014 $125.00	No	Yes	No		New Initial	[redacted]
MSU	[redacted]	12/15/2008	Pro(k)	3	$125.00	$125.00	Payment made 7/18/2014 $125.00	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	8/29/2014	Pro(k)	1	$50.00	$16.67		Yes	No	No		New Initial	[redacted]
MSU	[redacted]	4/30/2013	Pro(k)	3	$87.50	$87.50	Payment made 7/18/2014 $75.00	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	9/16/2011	Pro(k)	3	$125.00	$125.00	Payment made 7/18/2014 $125.00	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	12/13/2011	ProAdvice ONLY	3	$62.50	$62.50	Payment made 7/18/2014 $62.50	Yes	No	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	11/27/2011	ProAdvice ONLY	3	$62.50	$62.50	Payment made 7/18/2014 $62.50	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Update Order Status

Select	Size	Location	Item Name	Description	Status
☐	106" x 543"		106" x 543" Banner	13oz Vinyl: No Finishing	Shipped: 7/1/2014, by: Chelsea George, Shipper: UPS, Tracking#: 1Z6Y383A1544916915, Scheduled Delivery Date: 7/2/2014 PO 063014-4200 Assigned: 6/30/2014 1:37:47 PM, by: Chelsea George
☐	106" x 96"		106" x 96" Banner	13oz Vinyl: No Finishing	Shipped: 7/1/2014, by: Chelsea George, Shipper: UPS, Tracking#: 1Z6Y383A1544916915, Scheduled Delivery Date: 7/2/2014 PO 063014-4200 Assigned: 6/30/2014 1:37:47 PM, by: Chelsea George

Generate Shipping Labels

Instructions edit

Shipping Select the items you are shipping. Enter the **Shipper**, **Tracking #** and **Ship Date**. When finished click **Update Shipment Status** button.

Order Status Shipped Update Status

Purchase Order

PO#

Vendor

PO Amount

Due Date

Pick Up Date

Comments to Vendor

Update PO Status

Shipment

Shipper Select shipper

Tracking #

Ship Date

(optionally, include time as "hh:mm:ss")

Scheduled Delivery Date

Update Shipment Status

Delivery

Delivery Date

(optionally, include time as "hh:mm:ss")

Update Delivery Status

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

BBQTrophies.com

This site had been recently built with numerous flaws:

- First and foremost, for many customers, the shopping cart failed to work – you couldn't add an item to the cart.
- In that every trophy is customized with the name of the event, the nature of the trophy/prize and the winner's name, the 'customized details' field should have been required. Orders came in without the details meaning that BBQTrophies had to contact the customer to get it.
- Any orders totaling $200 or more should get free shipping, but this was not working.
- Checkout should offer a variety of UPS shipping options/rates, but this was not working.
- Managing the online store, including items offered and order management, was difficult.
- The entire cart was implemented using 'frames', so that a 3rd party shopping cart solution was embedded within the pages of the bbqtrophies.com site. While there are many reasons to avoid this technique, the most serious is that it destroys the SEO for the site. None of the product listings/details exist for bbqtrophies.com – rather for the 3rd party website.
- In addition to the website/store software issues, the web developer was slow to respond to requests, such that after a few months, the site was still incomplete.

Direct Knowledge rebuilt the site:

- Fixing all of the flaws in the original site.
- With software that is much easier to use, both for customers and site manager.
- Adding a key feature: to allow customers to upload a custom graphic to be added to select trophies.
- With the bulk of the work done in less than 10 days.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

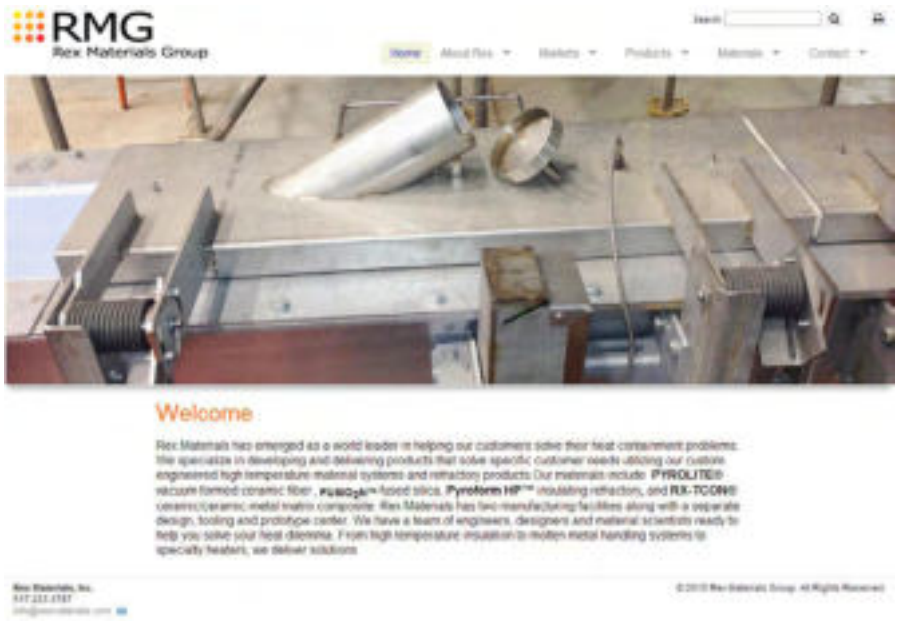
PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Rex Materials

Updated website to improve look, usability, navigation, SEO, and more.

- Redesigned this outdated website to look much more modern, ‘lighter’, wider, with site-wide primary navigation at the top, and section-specific sub-navigation in the left sidebar.
- Added larger, higher-quality photos and slideshows to better highlight RMG’s products and capabilities.
- Added a ‘call to action’ in the right sidebar to encourage contact.



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

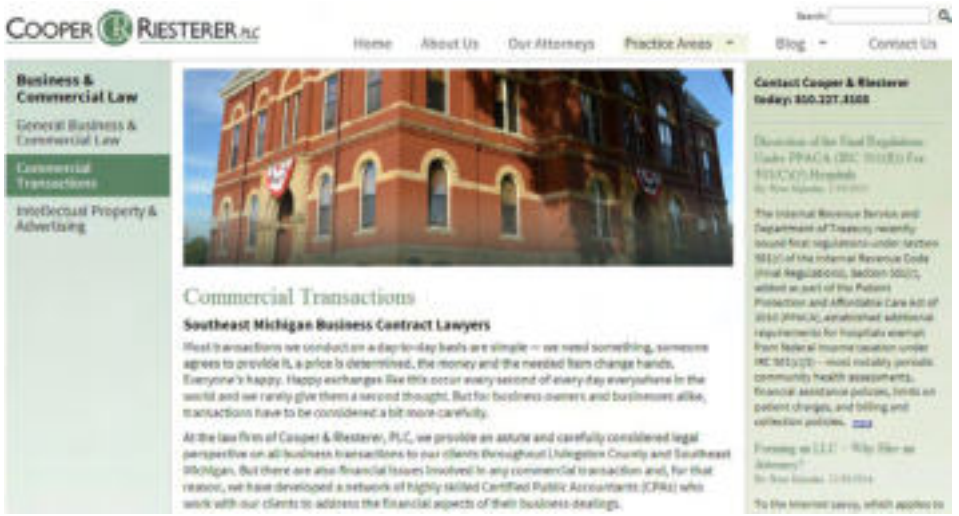
PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Cooper & Riesterer

Updated website to improve look, usability, navigation, SEO, and more.

- Redesigned look to be ‘lighter’, with site-wide primary navigation at the top, and section-specific sub-navigation in the left sidebar.
- Added right sidebar to include ‘call to action’ (phone number) and blog posts relevant to the main content on each page.
- Added mobile version of website.
- Reformatted blog pages to be easier to navigate and read.
- We’re still working on SEO updates.



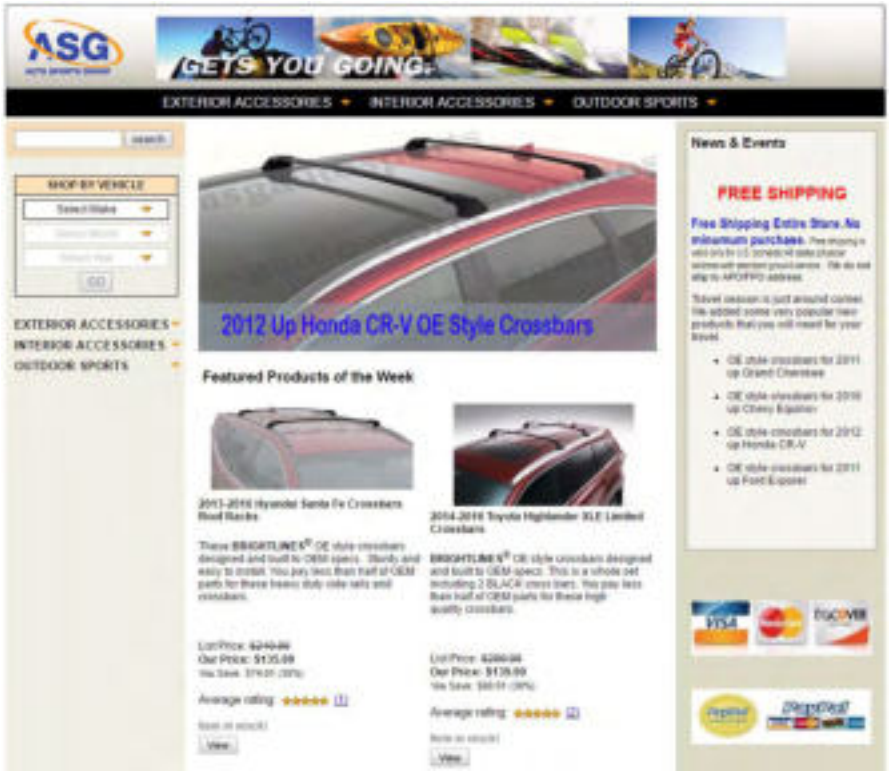
Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Auto Sports Group (asgautosports.com)

- DirectK built this ecommerce website including extensive customization to support ASG's business processes.
- $400,000+ annual sales via website
- Custom navigation options for auto make, model, and year selections.
- Custom shipping options via Fedex and USPS and for shipments to Canada and overseas.
- Integration with Shipworks a 3rd party shipping management solution.
- I will soon be working on an extensive SEO project to improve search engine rankings throughout the site.



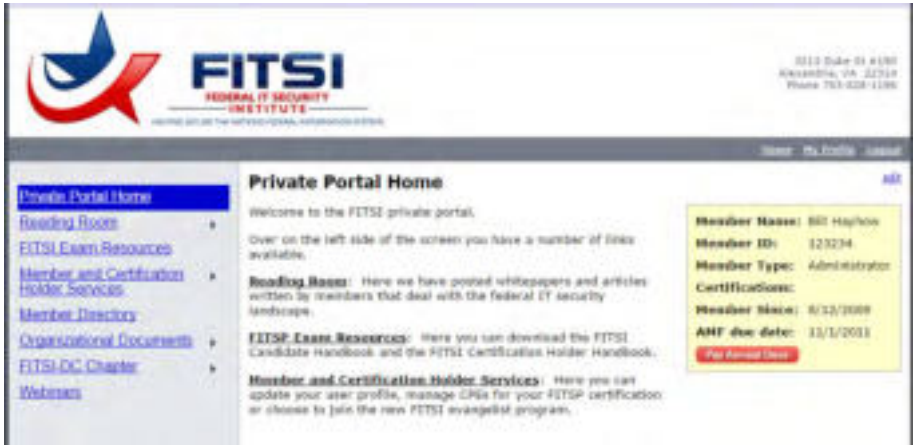
Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Federal IT Security Institute (fitsi.org)

- DirectK built the membership portal for FITSI including extensive membership management functionality.
- Built using Direct Knowledge's ActiveK Content Management System.
- Member functions include member directory, continuing education tracking, and access to a library of study and exam resources.
- Membership renewal management sends out notifications monthly prior to renewal dates and reminders monthly after renewal dates with online payment processing.
- FITSI provides member certifications, related to completion of specified training and exams, and the system provides functions for managing certifications and the certification process.
- Event management allows for multi-day, multi-session conferences and seminars, with separate registrations and fees for various sessions.



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Huron Valley Law Center (huronvalleylaw.com)

- DirectK redesigned this site to include improved layout, more/better photos, more content and marketing messages, better navigation and improved SEO.
- We recently added 100+ videos that HVLC had professionally produced, both to provide more personal answers to typical questions and also for SEO.
- Built using Direct Knowledge's ActiveK Content Management System, to allow HVLC staff to manage all of the content within the website.
- Extensive use of slideshows and photos with captions to present marketing messages.
- Events functionality to show and hide events based on the calendar.
- Blogs allow HVLC to post topics of interest to clients and for social media and SEO.

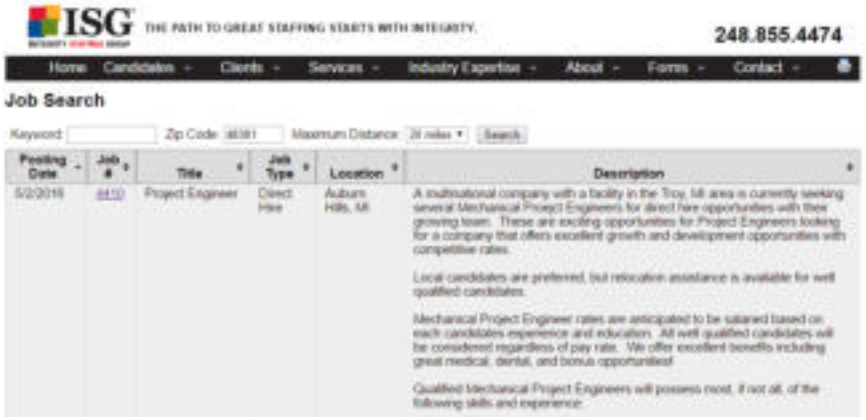
Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Integrity Staffing (isg-staffing.com)

- Built using Direct Knowledge's ActiveK Content Management System, to allow HVLC staff to manage all of the content within the website.
- Custom job management system allows ISG to post a variety of positions they are working to fill.
- Candidates can search for positions based on keyword and/or zip code (proximity).
- Candidates can submit resumes and contact information via online form.
- Placed candidates can download employment forms.



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

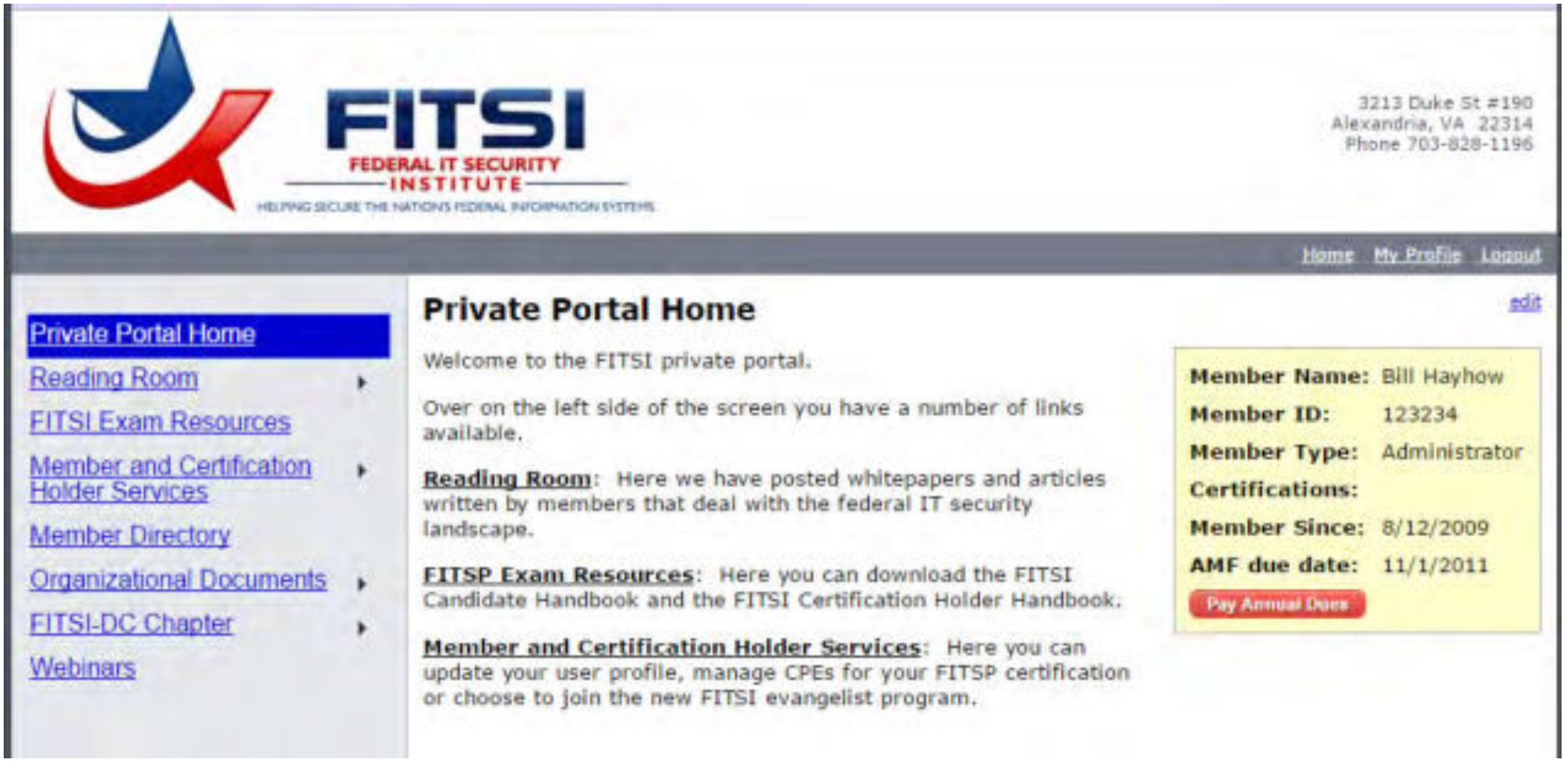
For Feets Sake (4feetssake.com)

- Built using Direct Knowledge's ActiveK Content Management System, to allow FFS staff to manage all of the content within the website.
- Custom home page design allows FFS to list a variety of products and events in easy-to-browse layout.
- Online store functionality allows FFS to offer shoes in a variety of styles, sizes, and colors.
- Custom checkout allows acceptance of orders with minimal customer information, to allow FFS to verify stock before completing the sale.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

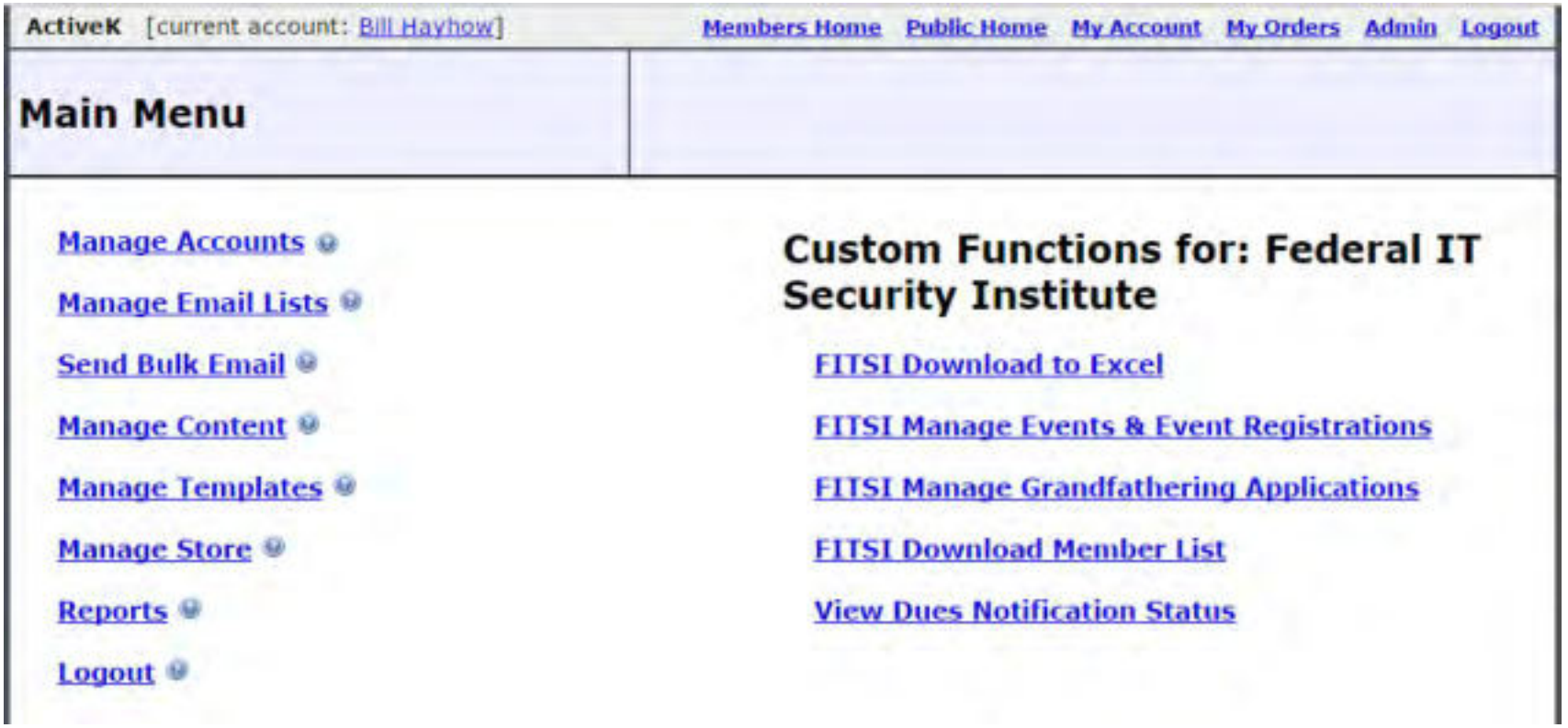
EMAIL: info@directK.com
WEB: www.directK.com



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

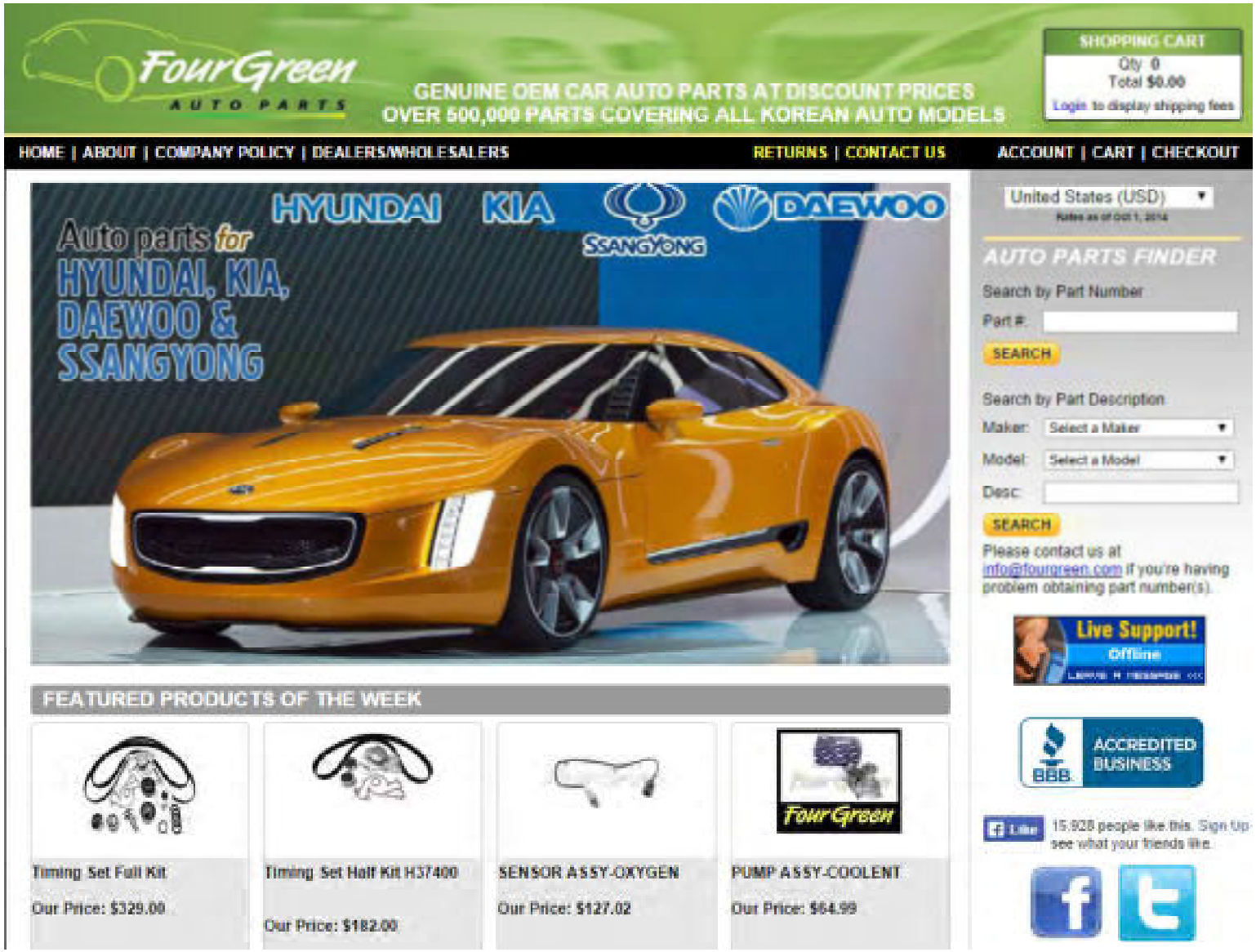


Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge



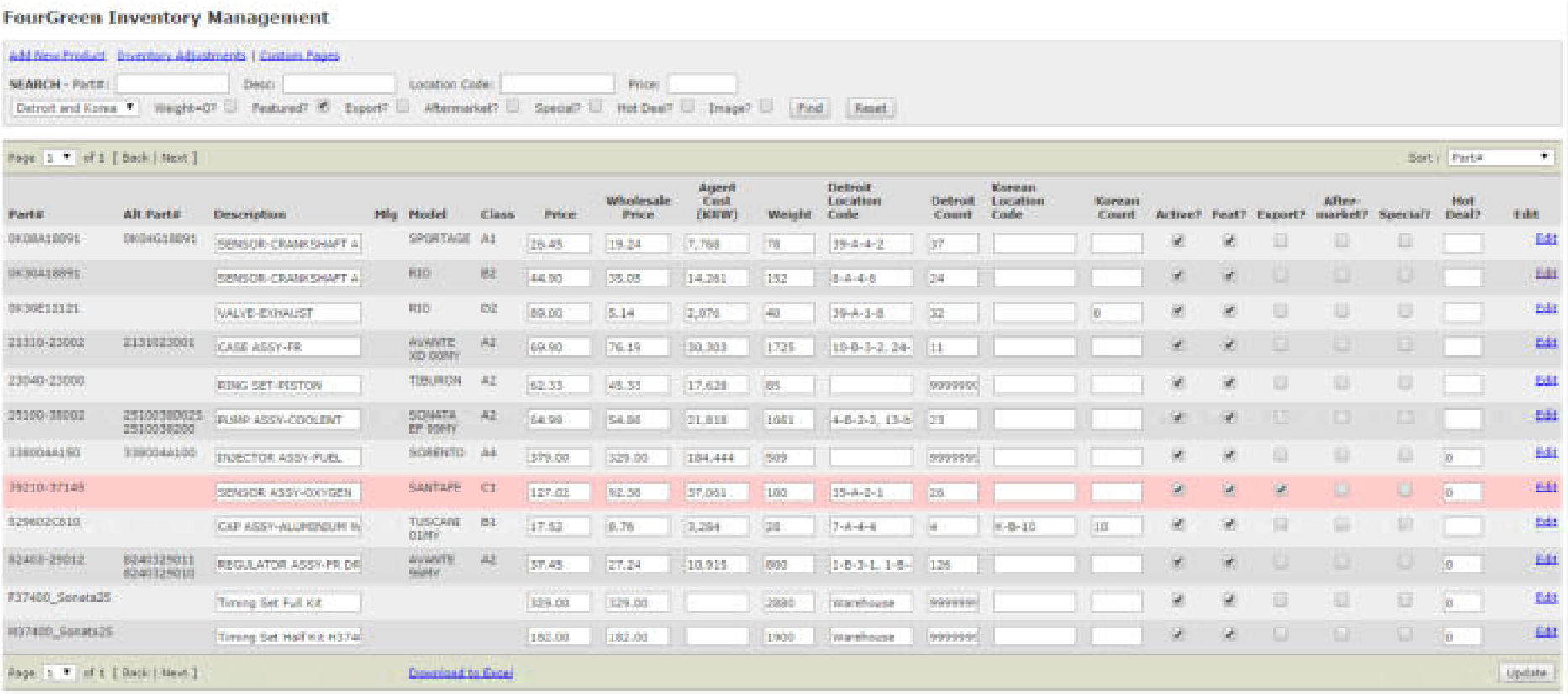
Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

FourGreen Inventory Management



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

StraightLine

Personal. Unbiased. Proactive.

Client Login

StraightLine was created over ten years ago to provide plan design assistance, participant education, advice and management to universities, colleges, corporations and individuals. Our mission is to ensure that each and every employee and individual have the tools necessary to reach their retirement goals.

Higher Education
Integrated Plan Services

Expert Fiduciary care for your plan and participants.

401(k) / 403(b)
Managed Accounts

How do SMART people manage their 401(k) / 403(b) accounts?

They don't.

STARS
Investment Services

You've never known an investment firm like us - - we always act in the best interest of our clients.

Wealth
Management

Put your assets where you put your trust.

© 2012 StraightLine. All rights reserved.

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

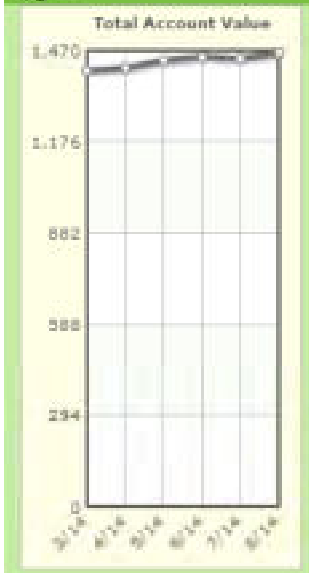
PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

directKnowledge

StraightLine
Personal. Unbiased. Proactive.

logout | Home | My Portfolio | My Communications | My Profile | Market Commentary | Contact Client Services



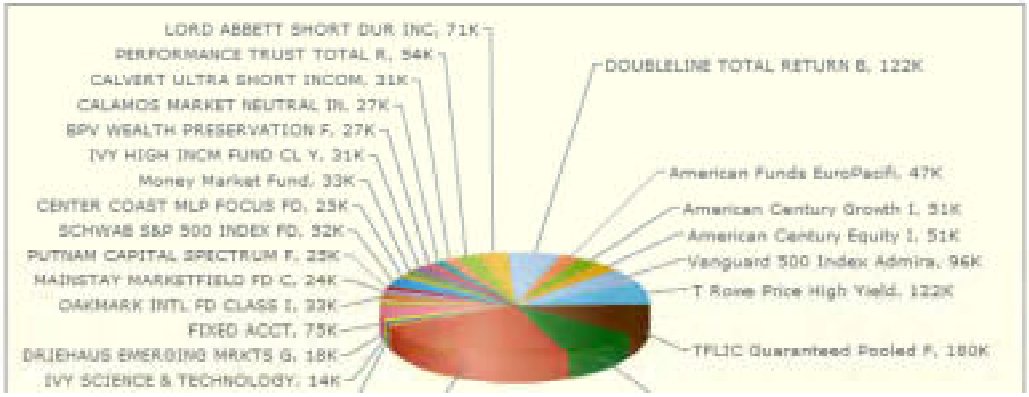
Portfolio Information

The information on these pages is gathered from your account custodian and is believed to be accurate, but cannot be guaranteed by StraightLine. We provide it as a service to you so you can see all of your account balances in one place.

In general the account data is from the previous business day's closing figures. On occasion the custodian or the program that gathers their information does not provide the data we need at the appointed hour and the results may be delayed for a day. There are a few custodians whose information cannot be updated daily and so values are only corrected at the end of each month.

If you find that your account data is not updating as you think it should, or if you have any other questions please give us a call at 866-401-5238.

Account Name	Account #	Type	Value
Fidelity NetBenefits - rollover	[illegible]	Pro (K)	$0.00
IRA	[illegible]	STARs - Schwab	$167,729.15
Lincoln Financial Group Annuities	[illegible]	Pro (K)	$75,268.33
TOD	[illegible]	STARs - Schwab	$467,468.58
Transamerica Retirement Solutions My TRSRetire	[illegible]	Pro (K)	$1,206,937.77
Total			**$1,917,403.83**



Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Billing - University Billing

« return to billing menu

Payments due as of: 9/30/2014

Generate payments in Salesforce

Export payment information to Excel

Plan	Client	PP Accept Date	Client Type	Months Invcd	Qtrly Amount Due	Pay Amount (prorated)	Most Recent Payment	Fid. elity	TIAA CREF	Van. guard	MSU 457	Comments	Account Details
NU	[redacted]	10/22/2013	Pro(k)	3	$125.00	$125.00	Payment made 7/18/2014 $125.00	No	Yes	No		New Initial	[redacted]
MSU	[redacted]	12/15/2008	Pro(k)	3	$125.00	$125.00	Payment made 7/18/2014 $125.00	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	8/29/2014	Pro(k)	1	$50.00	$16.67		Yes	No	No		New Initial	[redacted]
MSU	[redacted]	4/30/2013	Pro(k)	3	$87.50	$87.50	Payment made 7/18/2014 $75.00	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	9/16/2011	Pro(k)	3	$125.00	$125.00	Payment made 7/18/2014 $125.00	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	12/13/2011	ProAdvice ONLY	3	$62.50	$62.50	Payment made 7/18/2014 $62.50	Yes	No	No		Anniversary portfolio missing (has inactive account)	[redacted]
MSU	[redacted]	11/27/2011	ProAdvice ONLY	3	$62.50	$62.50	Payment made 7/18/2014 $62.50	No	Yes	No		Anniversary portfolio missing (has inactive account)	[redacted]

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Weaver Media, LLC.
Creative imaging solutions

Home | Select Facility | Orders | Inventory | WMOnline | Menu Boards / Banners | Contact Us | Logout

Manage Orders

Facility: Daytona International Speedway (6)

Order Search

Order #

Sponsor

Order Date Start

Order Date End

Order Status

Search

Instructions edit

If prompted for 'facility', please select a facility. Then, specify values for any fields available to limit the search to the items you want to view. Remember that less is more when using the **Keyword** search function.

To display all orders simply leave all fields blank and click **Search**.

View allows you to see the details of the order.

Search Results

download complete report to Excel | download brief report to Excel | download vendor report to Excel

Order #	Requested Delivery Date	Status	Description	Sponsor/	Order Contact	Vendor	
4200	7/2/2014	Shipped	Daytona Banners	Daytona		CGS Imaging	view
4160	6/24/2014	Shipped	Small Coke Trailers	Coca-Cola		CGS Imaging	view
4144	6/19/2014	Shipped	Coke Trailer Wraps	Coca-Cola		CGS Imaging	view
3578	1/13/2014	Shipped	Rolex 24 Adhesive	Rolex		CGS Imaging	view
2610	2/19/2013	Shipped	Daytona Ambulance Decals	Halifax		CGS Imaging	view
2602	2/20/2013	Shipped	Quicken Loans Car Topper	Quicken Loans		CGS Imaging	view

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com

Update Order Status

Select	Size	Location	Item Name	Description	Status
☐	106" x 543"		106" x 543" Banner	13oz Vinyl: No Finishing	Shipped: 7/1/2014, by: Chelsea George, Shipper: UPS, Tracking#: 1Z6Y383A1544916915, Scheduled Delivery Date: 7/2/2014 PO 063014-4200 Assigned: 6/30/2014 1:37:47 PM, by: Chelsea George
☐	106" x 96"		106" x 96" Banner	13oz Vinyl: No Finishing	Shipped: 7/1/2014, by: Chelsea George, Shipper: UPS, Tracking#: 1Z6Y383A1544916915, Scheduled Delivery Date: 7/2/2014 PO 063014-4200 Assigned: 6/30/2014 1:37:47 PM, by: Chelsea George

Generate Shipping Labels

Instructions edit

Shipping Select the items you are shipping. Enter the **Shipper**, **Tracking #** and **Ship Date**. When finished click **Update Shipment Status** button.

Order Status Shipped Update Status

Purchase Order

PO#

Vendor

PO Amount

Due Date

Pick Up Date

Comments to Vendor

Update PO Status

Shipment

Shipper Select shipper

Tracking #

Ship Date

(optionally, include time as "hh:mm:ss")

Scheduled Delivery Date

Update Shipment Status

Delivery

Delivery Date

(optionally, include time as "hh:mm:ss")

Update Delivery Status

Direct Knowledge, LLC.
510 Highland Ave., #295
Milford, MI 48381

PHONE: 248.478.9965

EMAIL: info@directK.com
WEB: www.directK.com